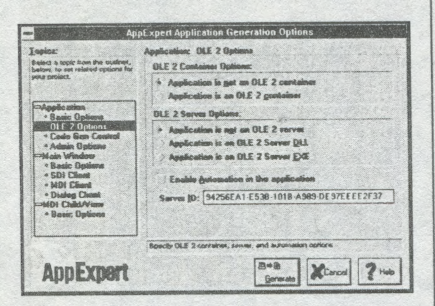
ISSN 0868-6157

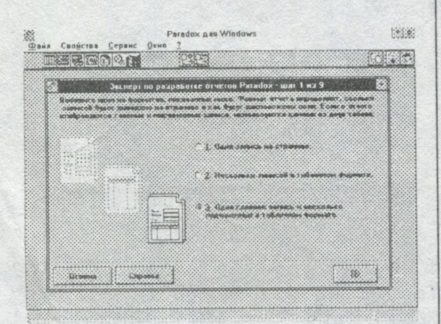
КОМПЬЮТЕР ПРЕСС

3'95



СПЕЦВЫПУСК «СТРАНА ПО ИМЕНИ BORLAND»





ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

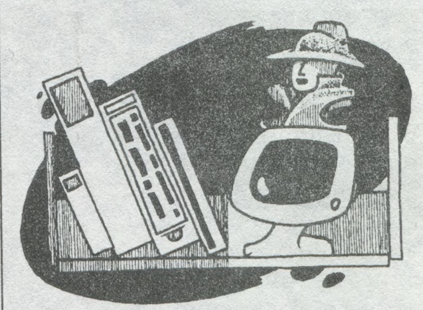
Пример 1 Поиск корня уравнения Решение с помощью команды *Подбор параметра...* в меню *Сервис*		
Начальные значение X и корень	*1,732161*	В ячейке B4 было 3, должно быть 1,732051...
Функция	0,000383	В ячейке B5 формула B4^2-3
Решение с помощью команды *Подбор параметра...* в меню *Сервис*		
Начальные значение X и корень	3	В ячейке B8 было 3, должно быть 1,732051...
Функция	0,000600	В ячейке B9 формула (B8^2-3)*0,0001
Решение с помощью команды *Поиск решения...* в меню *Сервис*		
Начальные значение X и корень	*1,730923*	В ячейке B12 было 3, должно быть 1,732051...
Функция	0,000000	В ячейке B13 формула (B12^2-3)*0,0001
Метод секущих		Точность 0,1 - корень 1,8 Точность 0,01 - корень 1,721529 Точность 0,001 - корень см. в B12



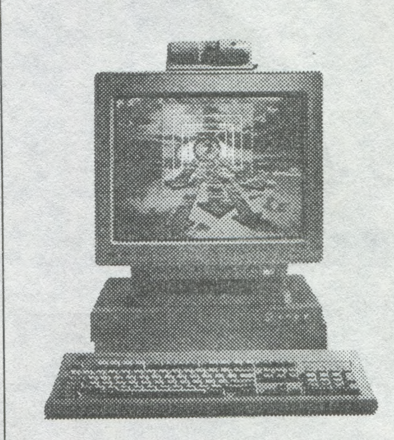
ВОПРОСЫ И ОТВЕТЫ



КНИЖНАЯ ПОЛКА



АППАРАТНОЕ ОБЕСПЕЧЕНИЕ



КОМПЬЮТЕР ПРЕСС




КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
3'95 (63)

Главный редактор:
Б.М.Молчанов

Редакционная коллегия:
К.С.Ахметов
А.Е.Борзенко
А.Е.Любимов
И.Б.Могучев
С.К.Новосельцев
А.В.Синев
(зам. главного редактора)
А.Г.Федоров

Литературный редактор:
Т.А.Шестернева

Корректор:
Т.И.Колесникова
М.М.Виноградова

Художник:
М.Н.Сафонов

Компьютерная верстка:
В.В.Голубков
О.В.Новокшонов

Ответственный секретарь:
Е.В.Кузнецова

Отдел распространения:
Т.В.Маркина

Отдел рекламы:
И.Ю.Борисов
Н.Н.Кузина
И.Б.Могучев
(зав. отделом)

Адрес редакции:
113093 Москва, а/я 37
Факс: (095) 470-31-05
Отдел распространения: (095) 471-32-63
Отдел рекламы: (095) 470-31-05
E-mail: editors@cpress.msk.su

РЕКЛАМА В НОМЕРЕ:

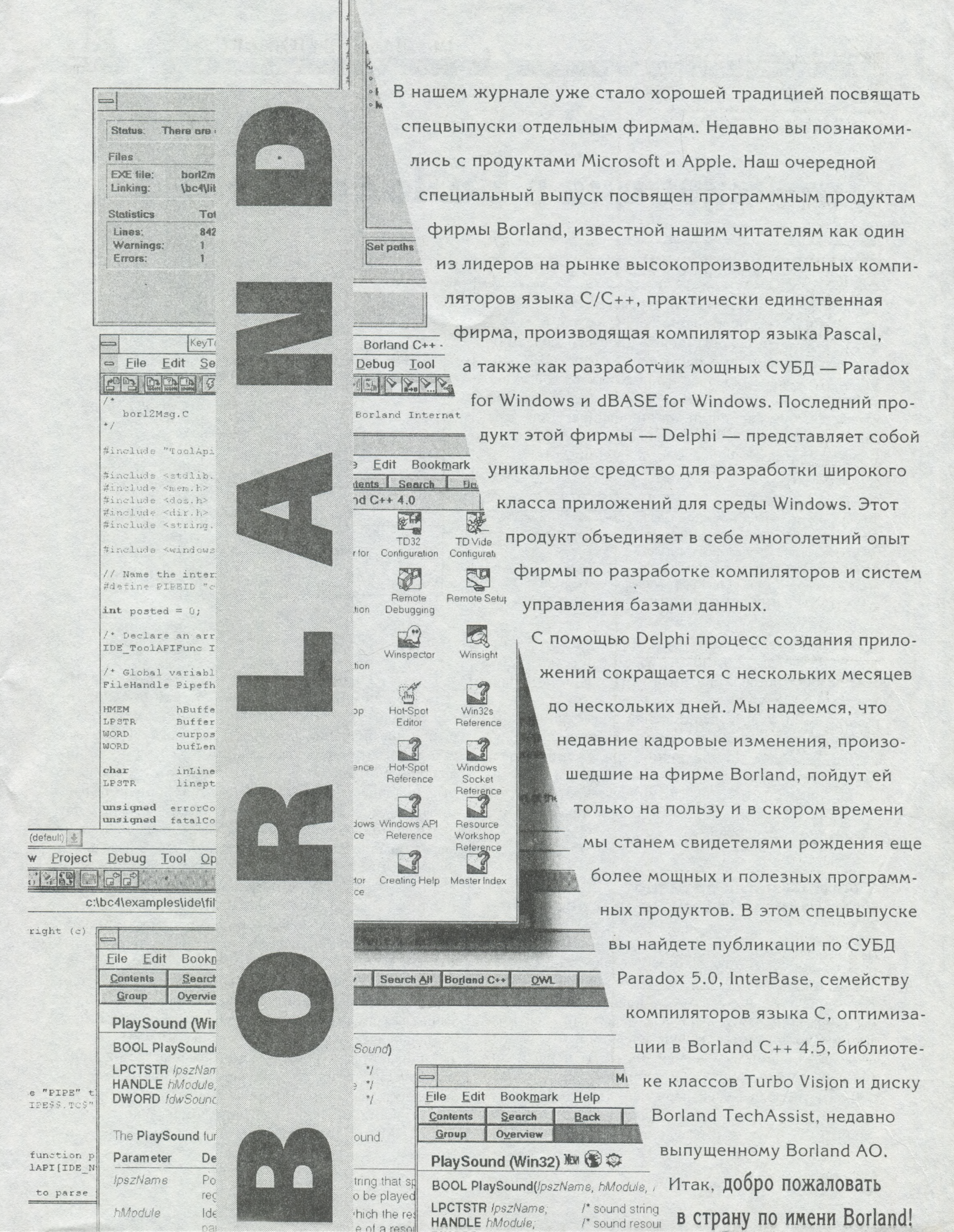
Ответственность за информацию, приведенную в рекламных материалах, несет рекламодатель

Сдано в набор 10.02.95. Подписано к печати 1.03.95. Формат 84x108/16.
Печать офсетная. Бумага офсетная. Усл.печ.листов 13,44+1,68 вкл.
Кр.-отт. 21,84. Тираж 52000 экз. (первый завод 30000). Заказ 532. С-39.

Оригинал-макет подготовлен фирмой «КомпьютерПресс».

Регистрационный № 400 от 5 октября 1990 г.

Отпечатано с оригинал-макета
в полиграфической фирме «КРАСНЫЙ ПРОЛЕТАРИЙ»
103473, Москва, Краснопролетарская, 16.



В нашем журнале уже стало хорошей традицией посвящать спецвыпуски отдельным фирмам. Недавно вы познакомились с продуктами Microsoft и Apple. Наш очередной специальный выпуск посвящен программным продуктам фирмы Borland, известной нашим читателям как один из лидеров на рынке высокопроизводительных компиляторов языка C/C++, практически единственная фирма, производящая компилятор языка Pascal, а также как разработчик мощных СУБД — Paradox for Windows и dBASE for Windows. Последний продукт этой фирмы — Delphi — представляет собой уникальное средство для разработки широкого класса приложений для среды Windows. Этот продукт объединяет в себе многолетний опыт фирмы по разработке компиляторов и систем управления базами данных.

С помощью Delphi процесс создания приложений сокращается с нескольких месяцев до нескольких дней. Мы надеемся, что недавние кадровые изменения, произошедшие на фирме Borland, пойдут ей только на пользу и в скором времени мы станем свидетелями рождения еще более мощных и полезных программных продуктов. В этом спецвыпуске вы найдете публикации по СУБД Paradox 5.0, InterBase, семейству компиляторов языка C, оптимизации в Borland C++ 4.5, библиотеке классов Turbo Vision и диску Borland TechAssist, недавно выпущенному Borland AO.

Итак, **добро пожаловать в страну по имени Borland!**

Оптимизация в Borland C++ 4.5

Алексей Федоров

В КомпьютерПресс №1'95 мы рассмотрели основные новинки компилятора Borland C++ 4.5, преимущественно возможность более простого создания приложений, использующих технологию OLE 2.0. Также кратко упоминалось о новых возможностях по оптимизации создаваемых программ. В этом обзоре мы рассмотрим ряд опций компоновщика TLink версии 7.0, входящего в комплект поставки Borland C++ 4.5. Удобнее всего собрать их в таблицу, что мы и сделаем:

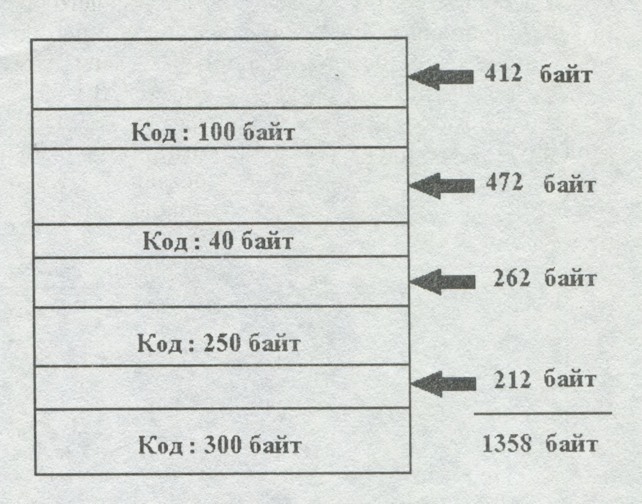
Опция	Тип оптимизации
/Gn	Пометить нерезидентную таблицу имен как выгружаемую
/Gr	Экспортировать резидентные имена по номерам
/Oc	Удаление "цепочечных" ссылок
/Oi	Сжатие сегментов данных
/Oa	Минимизация размера выравнивания сегментов
/Or	Минимизация размера выравнивания ресурсов

Отметим, что опции **/Gx** были доступны и в компиляторе версии 4.0, но мы рассмотрим их в данном обзоре, так как они имеют отношение к нашему разговору.

После того как мы перечислили новые опции компоновщика, давайте посмотрим, как они влияют на размер и скорость выполнения приложений (здесь необходимо отметить, что все новые типы оптимизации применимы только к Windows-программам и библиотекам). Начнем с минимизации размера выравнивания сегментов.

Минимизация размера выравнивания сегментов и ресурсов

Как известно, размер выравнивания сегментов задается опцией /A, после которой следует число, указывающее размер в байтах, кратный степеням двойки. По умолчанию используется значение 512 байт — размер сектора в MS-DOS. Что задает это значение? Давайте посмотрим на гипотетический исполняемый файл, состоящий из 4 сегментов. Пусть первый сегмент имеет размер 100 байт, второй — 40, третий — 250, а четвертый — 300. При использовании значения по умолчанию мы получаем размер сегмента, равный 512 байтам. В результате наш исполняемый файл занимает на 1358 байт больше, чем это реально необходимо.



Размещение сегментов кода

Полтора килобайта, скажете вы, это несерьезно. Но если мы вспомним, что Windows-программа может содержать до 255 сегментов, то 340 лишних байт (усредненное значение для нашего примера) на каждый сегмент — это достаточно много — 87 Кбайт. До появления Borland C++ 4.5 решением этой проблемы был подбор оптимального значения размера выравнивания сегментов (с помощью опции /A). Как показывает опыт, наиболее оптимальным значением выравнивания сегментов является 16. Посмотрите, что будет в этом случае с нашим файлом. Мы потеряем всего 22 байта. Так, с помощью опции **/Oa** компоновщик берет на себя поиск оптимального значения для выравнивания сегментов. Здесь необходимо сделать одно замечание. Размер по умолчанию (512 байт) выбран не только из-за того, что он равен размеру сектора. Как известно, ссылки на сегменты (в терминах секторов) хранятся в заголовке исполняемого файла как 2-байтовые величины. Таким образом, максимальный номер сектора может быть 65 535. Это ограничивает размер исполняемого файла. При использовании значения по умолчанию размер файла может достигать 32 Мбайт (65 535 байт*512), а при использовании значения выравнивания, равного 16, — всего 1 Мбайт (65 535 байт*16).

Минимизация размера выравнивания ресурсов

Новая опция компоновщика **/Or** позволяет выполнять минимизацию размера выравнивания ресурсов. Принцип здесь точно такой же, что и при минимизации размера сегментов. Можно рассматривать ресурсы просто как еще один тип сегмента — сегмент кода, сегмент данных и сегмент ресурсов. Интересно отметить, что компоновщик ресурсов в предыдущих версиях "следил" за действиями старшего брата — компоновщика исполняемых файлов и использовал то же значение для выравнивания сегментов.

Пора посмотреть на реальный пример. В качестве такого примера возьмем программу ACLOCK, входящую в комплект поставки компилятора. Размер исполняемого кода для нее составил 77 Кбайт (выполнялась компиляция без прилинковки стандартных библиотек).

Размер сегмента	Размер файла (байт)
16	71 408
32	71 680
64	72 064
512 (по умолчанию)	77 312
1024	83 968

Таким образом, становится очевидно: чем меньше размер выравнивания сегментов и ресурсов, тем лучше. Но при этом необходимо учитывать ограничения, накладываемые на размер исполняемого файла, о чем говорилось выше.

Упаковка области быстрой загрузки (Options|Project|Resources)

Хотя эта опция присутствовала и в версии 4.0, ее следует упомянуть в ходе нашего рассмотрения различных способов оптимизации Windows-программ. Для контрольного примера ACLOCK размер исполняемого файла без включения этой опции составил 77 Кбайт, а с включенной опцией — 76 Кбайт. Таким образом, мы сэкономили 1 Кбайт за счет более разумного размещения ресурсов и сегментов. С помощью опции Pack Fastload Area не происходит конкретной "упаковки" области быстрой загрузки (известной также под названием gangload area), как может показаться из названия. Просто компоновщик ресурсов (а именно он отвечает за то, каким будет исполняемый файл) делает следующее: располагает все сегменты данных, сегменты кода с атрибутом "невыгружаемый" и сегмент, содержащий точку входа, в одной непрерывной области, называемой областью быстрой загрузки. По умолчанию в эту область попадают и сегменты с атрибутом "принудительная загрузка — preload". Загрузчик исполняемых программ в ядре Windows (модуль Kernel) "знает" о существовании этой области (эта область используется начиная с версии 3.0) и загружает все ее содержимое в память при запуске программы. Причем для области быстрой загрузки отводится huge-сегмент, и копирование ее содержимого выполняется одним вызовом функции _hread (huge read). За счет этого сокращается число обращений к диску для загрузки сегментов. Таким образом, наиболее часто используемые сегменты кода постоянно находятся в памяти, и программа может работать несколько быстрее. Процесс назначения атрибутов сегментам кода чисто эмпирический, но можно воспользоваться утилитой HeapWalk. Сразу же после загрузки программы посмотрите, какие сегменты присутствуют в памяти. Эти сегменты — кандидаты на то, чтобы попасть в область быстрой загрузки. Далее можно посмотреть, что меняется в процессе работы программы — какие сегменты загружаются, выгружаются и остаются в памяти. Это даст вам картину того, какие сегменты должны иметь атрибут "невыгружаемый". После этого необходимо модифицировать содержимое .DEF-файла и перекомпилировать программу, не забыв про опцию Pack Fastload Area. Если вы хотите автоматизировать процесс слежения за поведением программы в памяти, то можете создать небольшую программу, использующую нотификационные возможности библиотеки ToolHelp. С ее помощью вы будете узнавать о загрузке и выгрузке сегментов. Но создание такой утилиты — тема отдельного разговора.

Экспортирование резидентных имен по номерам

Это еще один вид оптимизации. На самом деле экономия дискового пространства в данном случае будет минимальной, но вы немного сократите объем памяти, занимаемой программой. Если эта опция установлена, то всем экспортируемым именам автоматически присваиваются номера, что сокращает время на процесс поиска ядром экспортируемых функций во время выполнения программы и освобождает вас от необходимости задания номеров функций в .DEF-файлах. Кратко вспомним, как имена функций попадают в таблицу резидентных имен. По теории, вы должны пользоваться следующей конструкцией в .DEF-файле:

```
EXPORTS
ABOUTPROC        @2        RESIDENTNAME
```

то есть непосредственно указывать модификатор RESIDENTNAME, но чаще всего при описании функции вы указываете ключевое слово _export. Когда компоновщик собирает программу и находит в объектном файле (созданном компилятором) запись типа EXPDEF (результат использования ключевого слова _export), то он ищет соответствующее имя функ-

ции в секции EXPORTS и, если не находит его, имя функции помещается в таблицу резидентных имен.

▲ Пометить нерезидентную таблицу имен как выгружаемую

Этот тип оптимизации представляет собой еще один способ сокращения объема памяти, занимаемого программой. Нерезидентная таблица имен загружается ядром Windows (модуль Kernel) в процессе обработки вызова функции GetProcAddress. Если эта таблица помечена как выгружаемая, то занимаемая ею память освобождается сразу же после того, как отработала функция GetProcAddress. Для того чтобы установить эту опцию, необходимо воспользоваться командой **Discard Nonresident Name Table** в диалоговой панели, вызываемой командой Options|Project|Linker|16-bit Linker или опцией компоновщика **/Gn**.

▲ Удаление "цепочечных" ссылок

Что такое цепочечные ссылки? Предположим, в вашей программе несколько раз вызывается функция MessageBox. Для каждого вызова будет создан специальный код, занимающий 8 байт, из которых только два первых имеют значение для программы. При установленной опции "удаление цепочечных ссылок" вместо кода для каждого вызова функции будет создан только один блок кода, а все ссылки будут трансформированы в более компактные, вызывающие общую точку входа. Такой подход может существенно сократить размер исполняемого файла. Отметим, что этой возможностью также обладают компоновщики Microsoft LINK и OPTLINK.

▲ Заключение

Итак, мы рассмотрели некоторые типы оптимизации, доступные в новом компиляторе фирмы Borland. Это поможет вам сделать программы еще более эффективными. Следует отметить, что примерно 30% коммерческих приложений поставляется с отладочной информацией. С одной стороны, это помогает легче разобраться с тем, как устроена и работает программа, а с другой — оставляет впечатление, что программа не совсем готова к выпуску. Более того, использование библиотек классов существенно влияет на размер отладочной информации — в этом случае размер программы может достигать нескольких мегабайт! Воспользовавшись утилитой TDSTRIP, можно удалить отладочную информацию практически из любой программы, но лучше, если об этом позаботится сам разработчик. Так что, собирая приложение после успешной отладки, не забудьте отключить соответствующие опции.

Системы программирования на языке С++

Краткий обзор семейства продуктов

Кирилл Раннев

Эта статья представляет собой описание состояния дел на начало 1995 года в семействе компиляторов Borland C++/Turbo C++. Мы рассмотрим системы программирования:

- Borland C++ 4.5;
- Borland C++ 4.5 and Database Tools;
- Borland C++ 2.0 for OS/2;
- Turbo C++;

и ряд дополнительных продуктов, входящих в семейство С++ компании Borland:

- Power Pack for DOS;
- Visual Solutions Pack 1.1;
- Turbo Assembler;
- Brief;
- ForeHelp.

Во время подготовки статьи более простые и компактные компиляторы линии Turbo C++ были представлены двумя достаточно долго находящимися на рынке продуктами версий 3.* — Turbo C++ for DOS и Turbo C++ Visual Edition for Windows. К моменту появления статьи в печати ожидается выход Turbo C++ 4.5. Поэтому в разделе "Turbo" речь пойдет об этом продукте. Автор заранее приносит извинение за возможные неточности, так как речь идет о пока еще необъявленном продукте.

▲Borland C++ 4.5

Новая версия системы программирования Borland C++ 4.5 объявлена в декабре 1994 года и с начала января 1995 года появилась в России в свободной продаже. Версия 4.5 — версия "переходного" периода, когда разработчикам приходится вести разработку приложений для 16-битных операционных систем и в то же время готовиться к тому, что в скором времени на рынке начнут доминировать 32-битные ОС. Мир приложений также станет 32-битным. Эта версия обеспечивает разработку 16- и 32-разрядных приложений. Возможно создавать программы для DOS, Windows 3.1, Win32s, Windows NT и Windows 95 (для ранних версий этой ОС). Обеспечивается совместимость по коду для всех разновидностей Windows. Средство Target Expert позволяет получить приложения для любой из этих систем простым выбором в окне диалога. Интегрированная среда системы программирования работает также в любых ОС семейства Windows.

Визуальные средства программирования, базовая библиотека классов OWL, функции интегрированной среды разработки, спецификация языка, соответствующие ANSI-стандартам языков С и С++ (включая templates, обработку исключительных ситуаций, RTTI — информацию о типах во время выполнения программы), технология отладки "Just-in-Time" сохраняют преемственность с версиями 4.0/4.02. Остановимся на основных особенностях версии 4.5.

▲OCF — поддержка OLE 2.0

Возможность встраивать поддержку технологии OLE 2.0 в разрабатываемые Windows-приложения является наиболее интересной особенностью новой версии.

Кратко охарактеризуем OLE 2.0. Это встроенная в Windows технология интеграции приложений, позволяющая одному приложению использовать данные и функциональные возможности других программ. При этом приложению, использующему "чужие" данные, не требуется дополнительной информации о функциональных возможностях и другой специфики задачи, которая создала эти данные. С точки зрения разработчика OLE представляет собой

систему интерфейсов, протоколов и соглашений по организации взаимодействия программ. Базовыми понятиями технологии OLE являются:

- составные документы (compound documents), организующие среду хранения для встроенных и/или связанных объектов;
- связь и внедрение — два способа соединения данных разных задач в документе одного приложения за счет установки ссылок на объекты (связь) или непосредственного встраивания "чужих" данных в документ (внедрение). Оба способа обеспечивают иерархическую вложенность объектов;
- редактирование (активация) по месту — доступ к функциям приложения, породившего встроенный объект, при обращении к встроенному объекту непосредственно на месте, в основном приложении, без переключения текущего окна;
- операции с Буфером обмена Windows и drag-and-drop-операции при редактировании объектов;
- автоматизация, позволяющая одному приложению "издавать" команды, которые влияют на поведение другой программы;
- OLE-клиент и OLE-сервер — характеризующие роли приложений при их взаимодействии. Клиент (иногда называемый контейнером) — это приложение, в документ которого встроены данные, обрабатываемые OLE-сервером. Многие, но не все приложения могут выступать в обеих этих ролях. Например, в категории баз данных Paradox 5.0 умеет работать как клиент и как сервер OLE 2.0. В качестве OLE-клиента Paradox умеет хранить и обрабатывать OLE-объекты в своих базах данных, в качестве OLE-сервера — обрабатывать базы данных, встроенные в документы других приложений. dBASE 5.0 for Windows и Access 2.0 выполняют функции только OLE-клиентов.

Технология интеграции приложений OLE 2.0 дает существенные преимущества и расширяет возможности для пользователей приложений, поддерживающих эту технологию. С точки зрения разработчиков она далеко не всегда вызывает положительные эмоции, хотя за счет OLE 2.0 достигается существенная экономия разработки: можно не разрабатывать некоторые функции, а поручить выполнение этих функций приложениям, в которых они уже реализованы.

Одним из источников неприятностей для разработчика является существенная сложность API (интерфейса прикладных программ) для доступа к этой технологии. Для включения поддержки OLE 2.0 приходится добавлять тысячи строк кода. Разработчики систем программирования в целях борьбы с этими сложностями обычно предлагают высокоуровневые надстройки, реализуемые, как правило, в виде библиотек классов, закрывающие сложности OLE 2.0 API. Такие возможности включены, например, в библиотеку классов MFC 2.5 компании Microsoft, входящую в ее компилятор Visual C++ 1.5 (и в более поздние версии). Компания Borland включила библиотеку OCF (Objects Components Framework) в компилятор Borland C++ 4.5.

Другим существенным источником трудностей является то, что в основе технологии OLE 2.0 лежит модель программирования, основанная на Component Object Model (COM), которая существенно отличается от привычной модели Windows-программирования, основывающегося на управляемой событиями (event-driven) структуризации алгоритма программы. Разработчик, только-только перестроившийся со стиля программирования, свойственного DOS, на стиль, принятый в Windows, должен осилить еще одну перестройку, связанную с Component Object Model. Кроме того, встраивание поддержки OLE в уже написанный и отлаженный Windows-код приводит к необходимости переработки структуры программы.

Технология поддержки OLE 2.0 компании Microsoft предполагает "революционный" способ решения этой проблемы. Разработчику предлагается создать новую структуру программы, соответствующую Component Object Model, а затем переносить в нее части кода старой программы, реализующего конкретные функции приложения. Это здорово, но скорее всего приведет к необходимости повторной отладки.

Библиотека классов OCF компании Borland позволяет трансформировать приложение в новое, имеющее поддержку OLE, без переработки его структуры. Собственно, это свойство самой OCF, которая работает не только с программами, написанными на Borland C++, то есть использующими базовую библиотеку классов OWL (ObjectWindows Library), но и допускает использование с другими базовыми библиотеками классов, например Microsoft Foundation Classes, или даже вообще без базовых библиотек классов — на "чистом" C++. Легкое превращение существующей Windows-программы в OLE-приложение обеспечивают соответствующие классы, включенные в новую версию базовой библиотеки классов компании Borland, — OWL 2.5.

В целом библиотека OCF как средство создания OLE-приложений обладает следующими особенностями:

- независимость от выбора компилятора языка C++ и поддержка различных базовых библиотек классов;
- обеспечение существенно более высокого уровня абстракции по сравнению с уровнем OLE 2.0 API;
- включены классы для полной реализация всех функциональных особенностей OLE-технологии: операции с Clipboard, drag-and-drop-операции, элементы пользовательского интерфейса, свойственного OLE-приложению, такие как диалоги "Insert Object", "Paste Special", "Paste Link", меню "правой кнопки мыши";

- когда OLE-клиент запрашивает OLE-сервер с целью получить доступ к объекту, сервер должен предоставлять и данные, и способ их отображения на экране. OCF обеспечивает разработчика классами для представления данных встраиваемого объекта и образа, необходимого для отображения объекта на экране (view/doc model);
- наличие класса TOcStorage, реализующего поддержку составных документов с точки зрения операций ввода/вывода;
- возможность превращения приложений в OLE-клиенты и OLE-серверы без переработки структуры программы (принципиальное отличие от других OLE-инструментов);
- возможность создания серверов автоматизации (automation servers) как в форме EXE, так и DLL. Последнее обеспечивает повышение быстродействие примерно в 4 раза по сравнению с EXE-серверами. EXE-серверы позволяют обеспечивать обмен между 16- и 32-битными приложениями;
- локализация; OLE-сервер должен иметь возможность "говорить" на том же языке, что и приложение-клиент (речь идет не об алгоритмических, а о человеческих языках), что важно для приложений, работающих на рынках нескольких стран. OCF позволяет при регистрации определить язык клиента и создавать приложения, имеющие многоязыковую поддержку.

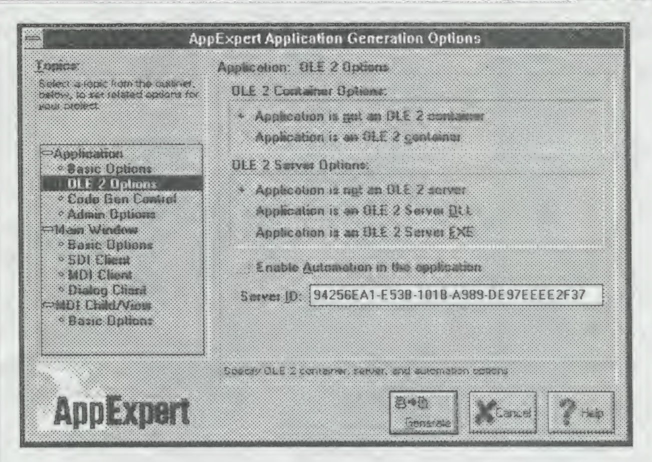
Настоящая версия поддерживает 16-битную реализацию OLE. В поставляемую версию компилятора включена предварительная версия 32-битного OLE. Официальная реализация будет включена в одну из ожидаемых в ближайшее время Update-версий. 16- и 32-битные реализации совпадают с точки зрения исходного текста разрабатываемой задачи, и переход между ними потребует лишь перекомпилировать программу.

OWL 2.5 и средства визуального программирования

Будучи независимой от реализации языка C++, OCF все же дает преимущества тем разработчикам, которые используют базовую библиотеку OWL и средства визуального программирования Borland C++ 4.5.

Визуальный AppExpert, появившийся в версии 4.0, обеспечивал создание всех необходимых компонентов Windows-приложения: от структуры исходных текстов и ресурсов и до RTF-файлов для построения справочных подсистем.

В версии Borland C++ 4.5 этот эксперт позволяет также указать перечень параметров приложения по поддержке технологии OLE как для вновь создаваемого приложения, так и для приложения, созданного ранее с помощью AppExpert. Это достигается



за счет специальных классов, TOleDocument и TOleWindow, включенных в новую версию OWL 2.5 и поддерживаемых из AppExpert и ClassExpert. ClassExpert позволяет придать соответствующие функции приложению, созданному с помощью AppExpert, за счет соответствующей настройки базовых классов этого приложения и добавления новых классов. С помощью ClassExpert обеспечивается добавление поддержки OLE-автоматизации к практически к любым классам.

VBX в 16- и 32-разрядных приложениях

Для разработчиков 16-битных приложений стало достаточно широко распространенным и удобным приемом разработки использование в приложениях визуальных компонент, реализующих различные прикладные функции и подготовленных в формате VBX (Visual Basic Extension). Сформировалось даже некоторое направление в индустрии программного обеспечения, когда крупные фирмы-производители, а главное, большое количество небольших независимых поставщиков стали поставлять на рынок комплекты VBX различной прикладной направленности. К сожалению, VBX в том виде, как они есть, являются объектами 16-битного мира. В сообщество 32-разрядных ОС и приложений их путь лежит через некоторую процедуру конвертации, для которой необходим их исходный код. Если его нет, то увы...

Borland C++ 4.5 включает уникальный механизм VBX32 для поддержки 16-битных VBX в 32-разрядных приложениях без каких бы то ни было дополнительных усилий со стороны разработчика. Для того чтобы превратить 16-битную программу, использующую VBX, в 32-битную, нужно только перекомпилировать ее код. С самим же VBX ничего делать не придется.

В будущих Update-версиях Borland C++ 4.5 будет включена поддержка OCX (OLE Custom Extension).

▲Новый оптимизатор и генератор кода

Изменения в генераторе кода, оптимизаторе и компоновщике привели к ускорению построения задачи, более быстрой работе кода и компактности получаемого модуля. 16-битные приложения получили новый компоновщик, 32-битные — новые компилятор и оптимизатор.

16-битный компоновщик содержит набор средств, которые после завершения построения задачи проводят послекомпоновочный процесс оптимизации или "вычищения" получившегося EXE- или DLL-файла. Это приводит к сокращению размера результирующего файла в среднем на 15% и снижению времени запуска программы примерно на 10%.

Другим способом сокращения размера исполняемого файла является возможность явного отключения средств обработки исключительных ситуаций в задачах, где эти возможности не используются (и не используются библиотеки, в которых есть обработка исключительных ситуаций). Это приводит к сокращению overhead исполняемых модулей, особенно значительному в маленьких задачах.

В 32-битном компиляторе используется новый генератор кода и оптимизатор. Генерация кода выполняется примерно вдвое быстрее, чем в предыдущей версии. При построении кода без оптимизации по скорости получающийся код работает примерно на 25% быстрее, чем в предыдущей версии. Для оптимизированного кода выигрыш также в пользу новой версии — примерно на 7-10%.

Технология OpenHelp ▲и новая документация

В интегрированную среду Borland C++ 4.5 добавлена новая технология динамического подключения файлов справки, называемая OpenHelp. Таким образом, у пользователя появляется возможность, находясь в интегрированной среде разработки Borland C++ 4.5, получать подсказку по всему набору используемых документов, включая библиотеки и утилиты Borland и других фирм. Поиск по ключевым словам ведется по всему набору активных справочных файлов.

У пользователей, имеющих компилятор Borland C++ 4.5 на CD-ROM, в дополнение к системе подсказок имеется полный комплект on-line-документации, помещенный в гипертекстовую систему поиска и просмотра *Dyna*Text TM. При этом формат этой документации не повторяет формат бумажной, он специально организован для оптимального размещения на экране. Поиск расширен возможностями использования словосочетаний, шаблонов, неточного поиска и задания логических выражений. Возможен перенос фрагментов документации в файлы пользователя. Большинство поставляемых в Россию комплектов Borland C++ 4.5 имеют также бумажный комплект документации. Вообще говоря, наличие книг в комплекте документации следует оговаривать в момент покупки.

Содержание документации также значительно переработано и расширено. В дополнение к справочной системе и on-line-документации изучению языка C++ помогает большое количество примеров, а также полные исходные тексты утилит, Runtime-библиотеки и библиотек классов OWL и OCF.

▲Объявленные Update-версии

Достаточно нетипичным событием является объявление компанией Borland планируемых Update одновременно с выпуском самой версии Borland C++ 4.5. Версия Update 1 запланирована на первую половину 1995 года и будет содержать официальную версию 32-битного OLE 2.0 и инструментарий Pipeline Registration. Выход версии Update 2 в основном зависит от компании Microsoft и ее планов выпуска Windows 95. Версию Update 2 планируется выпустить практически одновременно с этой операционной системой. Основное назначение этой версии — привести файлы заголовков и утилиты в соответствие с официальной версией Windows 95. (Соответствие версии Borland C++ 4.5 ранним версиям Windows 95 и возможность работать в них имеется уже сейчас.)

Предполагается, что Update-версии будут доступны легальным пользователям Borland C++ 4.5 по цене носителя и стоимости доставки.

▲Требования к ресурсам

Borland C++ 4.5 требует процессор 80386 или выше, 8 Мбайт памяти, DOS 4.01 или выше, Windows 3.1 или выше, мышь, дисплей не хуже VGA. 16-битные приложения можно создавать и на 4 Мбайтах.

Минимальная установка (только утилиты и компилятор, вызываемый из командной строки, минимум библиотек) требует 25 Мбайт дискового пространства, установка для работы под Windows — 75 Мбайт, полная — 100 Мбайт. Полная установка при использовании CD-ROM — 25 Мбайт. Комплект поставки на дискетах — 28 дискет.

▲Borland C++ 4.5 and Database Tools

Этот продукт представляет собой комплект для разработчика приложений, работающих с базами данных. В него входят полный компилятор Borland C++ 4.5, ядро процессора баз данных Borland Database Engine 2.0, набор драйверов для доступа к серверам баз данных Borland SQL Links 2.0, а также средство интерактивной работы с базами данных Borland Database Desktop 5.0.

Процессор баз данных Borland Database Engine 2.0, выпущенный в виде открытого средства разработки в октябре 1994 года, предоставляет разработчикам программного обеспечения в области баз данных доступ к мощным возможностям интерфейса IDAPI. Именно это ядро используется в широко известных системах управления базами данных компании Borland dBASE 5.0 for Windows и Paradox 5.0 for Windows. BDE ориентирован на разработчиков, создающих приложения, работающие с локальными базами, базами данных в локальной сети или в архитектуре "клиент/сервер".

BDE обеспечивает непосредственный доступ к данным форматов dBASE, Paradox и текстовым данным, через драйверы SQL Links 2.0 — к серверам баз данных Borland InterBase, Sybase, Oracle, Informix, MS SQL Server, через встроенную поддержку ODBC-драйверов (технология Idaptor) — к ODBC-данным. Idaptor транслирует вызовы функций интерфейса IDAPI в вызовы стандартных методов интерфейса ODBC. Это позволяет использовать практически любой драйвер стандарта ODBC в режиме IDAPI-драйвера. Технология Idapter существенно увеличивает число доступных через Database Engine форматов данных, позволяя подключать любой драйвер стандарта ODBC, поставляемый компанией Borland либо другой фирмой.

Технология IDAPI позволяет разработчикам реализовать доступ практически ко всем основным форматам баз данных в рамках одного приложения. Один и тот же код может поддерживать работу с несколькими форматами данных. В Borland Database Engine реализована возможность одновременного многопользовательского доступа к базам данных различных форматов.

Приложения, разрабатываемые с использованием Borland Database Engine, могут использовать для формирования запросов как язык SQL, так и режим "запроса по образцу" через язык запросов QBE. Процессор запросов работает с любым поддерживаемым форматом данных, включая SQL-запросы по данным в форматах Paradox и dBASE, при этом все необходимые средства для выполнения SQL- и QBE-запросов включены в BDE. Программа на C++, подготовленная с помощью Borland C++ 4.5 and Database Tools, может



работать с данными на уровне SQL- или QBE-предложений.

Кроме высокоуровневого SQL-доступа к данным Paradox и dBASE возможен процедурный, навигационный доступ. С помощью навигационного доступа в технологию IDAPI могут быть легко перенесены приложения, разработанные ранее на основе индексно-последовательного метода доступа (ISAM).

Для однотипной работы со столь разнообразными источниками данных и поддержки различных технологий их обработки введен некоторый уровень абстрагирования, связанный с концепцией курсоров. При этом поддерживаются связанные курсоры, двунаправленные курсоры, закладки, фильтры.

В комплект поставки Borland Database Engine включены все необходимые драйверы для поддержки национальных языков и расширенного набора символов, включая кодовые страницы 866 и 1251 для русского языка в кодировках DOS и Windows соответственно.

BDE обеспечивает высокую производительность приложений не только за счет оптимизации выполнения запросов и доступа к данным. В этот продукт включены средства, свойственные процессорам баз данных высокопроизводительных современных СУБД для Windows dBASE и Paradox, — буферизация ввода/вывода и кэширования. То есть BDE автоматически, прозрачно для программиста берет на себя работу по переносу работы с базами данных в оперативную память, минимизируя число дисковых операций ввода/вывода.

С точки зрения разработчика BDE представляет собой библиотеку из более чем 200 функций, покрывающую практически все аспекты работы с локальными базами данных, а также базами данных в локальных сетях и в приложениях с архитектурой "клиент/сервер". Создаваемые с помощью BDE приложения могут распространяться разработчиком, лицензированию подлежит только набор драйверов для доступа к серверам баз данных SQL Links.

Требования к системе те же, что и Borland C++ 4.5, только полная инсталляция требует 125 Мбайт оперативной памяти, и 8 Мбайт оперативной памяти являются обязательным минимумом.

▲Turbo C++ 4.5

Turbo C++ 4.5 представляет собой облегченную версию профессионального компилятора Borland C++ 4.5, предназначенную для обучения программированию на C++ для Windows или использования в тех приложениях, где полная мощность Borland C++ не требуется. Компилятор позволяет создавать 16-битные приложения для Windows 3.1. Интегрированная среда, характеристики языка, библиотеки OWL и OCF те же, что и в системе Borland C++ 4.5. Подробнее соотношение между Borland C++ 4.5 и Turbo C++ 4.5 может быть представлено таблицей:

	Turbo C++ 4.5	Borland C++ 4.5
Совместимость со стандартом ANSI	✔	✔
Полная реализация C++	✔	✔
Интегрированная среда разработки	✔	✔
Визуальные AppExpert, ClassExpert, TargetExpert, DialogExpert	✔	✔
Библиотека EasyWin, позволяющая программам для DOS и UNIX работать под Windows	✔	✔
Встроенная поддержка VBX	✔	✔
Интегрированный отладчик	✔	✔
Создание приложений, поддерживающих OLE 2.0	✔	✔
Создание приложений для Windows 3.1	✔	✔
Создание 32-битных приложений для Windows		✔
Создание приложений для DOS		✔
Turbo Debugger (автономный отладчик)		✔
Turbo Profiler		✔
Оптимизаторы		✔
Исходные тексты Runtime-, OWL-, OCF-библиотек		✔
Компилятор и утилиты, запускаемые из командной строки		✔

▲Borland C++ 2.0 for OS/2

Новая версия компилятора Borland C++ 2.0 для OS/2 была объявлена в декабре 1994 года вскоре после выхода Borland C++ 4.5. Эта версия является существенным шагом компании в направлении переноса технологии систем программирования Borland на различные системные платформы.

Основной особенностью этой версии является включение в нее библиотеки OWL для API Presentation Manager. Это даст уникальную возможность разделять и переносить исходный код между Windows и OS/2. Библиотека OWL для OS/2 практически полностью совместима с OWL 2.0 компиляторов Windows-линии, а от версии OWL 2.5 ее отличает отсутствие поддержки OLE в OS/2.

Кроме включения OWL для PM характерными чертами Borland C++ для OS/2 являются:
- поддержка современных стандартов языка C++, включая templates, обработку исключительных ситуаций и RTTI и классов для ANSI-строк;
- 32-битный оптимизирующий компилятор;
- интегрированная среда разработки для Presentation Manager;



- Resource Workshop для OS/2 и набор классов BPMCC (Borland Presentation Manager Custom Controls) — набор компонент для построения диалогового интерфейса;
- ObjectBrowser для просмотра иерархии объектов;
- Turbo Debugger и Turbo Assembler.

▲PowerPack for DOS

Borland PowerPack for DOS представляет собой набор инструментальных средств и библиотек для разработки 16- и 32-х разрядных DOS-приложений для защищенного режима процессора с использованием Borland C++ 4.02 и 4.5. Этот продукт рассчитан только на пользователей этих систем программирования. PowerPack for DOS включает 16- и 32-разрядные расширители DOS, новую версию библиотеки TurboVision и новую 32-разрядную версию библиотеки графического интерфейса BGI.

Для разработчиков DOS-приложений этот продукт предоставляет расширенные возможности защищенного режима процессора и линейной (FLAT) модели организации памяти программ, сочетая это с возможностью свободной дистрибуции разработанных приложений. 16- и 32-разрядные расширители DOS, поставляемые с пакетом, позволяют преодолеть барьер в 640 Кбайт используемой памяти, увеличивая объем доступной для программы виртуальной памяти до 4 Гбайт. Эти расширители DOS при установке пакета автоматически интегрируются в среду разработки (IDE) Borland C++.

Менеджер виртуальной памяти позволяет приложениям использовать больший объем оперативной памяти, чем реально установлено на компьютере. Для этого используется дисковое пространство, что происходит прозрачно для выполняемой программы.

Продукт также содержит сервисные средства для тех разработчиков, которые только начинают перенос своих приложений с платформы DOS в среду Windows или вынуждены одновременно поддерживать наборы исходного кода приложений для DOS и для Windows. Технология использования динамических библиотек (DLL) позволяет разработчикам применять одни и те же коды библиотек для DOS- и Windows-приложений. Это позволяет достигать большей мобильности и компактности программ для DOS и Windows, а также отлаживать как DOS-приложения защищенного режима, так и Windows-программы, даже используя тот же самый программный код.

Среди других особенностей — совместимость с библиотекой Borland Paradox Engine, 32-разрядная версия BGI, автоматическая интеграция в среду разработки (IDE) Borland C++ при установке продукта, поддержка отладчика TurboDebugger, а также новые 16- и 32-разрядные версии библиотеки TurboVision для защищенного режима DOS.

▲Turbo Assembler

Очевидным назначением этого продукта является изготовление наиболее критичных по скорости или памяти частей приложения. Turbo Assembler обеспечивает совместную разработку с C, C++ и другими языками программирования или же изготовление приложений исключительно на языке ассемблера. Возможно создание 16- и 32-битных приложений для DOS и Windows. В продукт также включен отладчик — Turbo Debugger.

В первой половине 1995 года ожидается выход новой версии этого продукта, содержащей доработки, оптимизирующие его использование вместе с новой системой программирования Delphi.

▲Visual Solutions Pack

Visual Solutions Pack 1.1 представляет собой набор из более чем 25 расширеннных управляющих элементов VBX. Расширенные управляющие элементы VBX первоначально были предложены как расширение программ на Visual Basic (Visual Basic eXtensions). Данный стандарт на диалоговые элементы стал популярен и в настоящее время множество разработчиков предлагают разнообразнейшие наборы VBX.



Поддержку VBX со стороны системы программирования Borland C++ обеспечивает TVBXControl, появившийся в OWL 2.0 и позволяющий управлять элементами VBX непосредственно из программ на C++.

Visual Solutions Pack 1.1 содержит следующий набор компонент:
- *Компоненту управления БД*, работающую с таблицами форматов dBASE IV и Paradox 3.5;
- *Электронную таблицу* неограниченного размера. Возможны: ввод формул, работа со шрифтами, использование более 120 встроенных функций;
- *Блокнотные "корешки"*, организующие электронные таблицы или другие приложения в трехмерный блокнот;
- *Текстовый процессор, реализующий режим WYSIWYG* и поддерживающий работу со стилями, шрифтами и т.д.;

- *Коммуникационный интерфейс,* позволяющий передавать файлы и эмулировать терминалы ANSI и TTY;
- *Редактор графических файлов;*
- *Свыше 30 трех- и двумерных типов диаграмм*;
- *Набор средств украшения интерфейса*: "живые" кнопки, индикаторы, вращатели, ползунки, часы, звонки и даже игральные кости и карты.

При добавлении каждого нового расширенного элемента в Resource Workshop в палитру диалоговых элементов ToolsPalette добавляется соответствующая пиктограмма. После этого VBX становится доступен для использования в Workshop.

В версию 1.1 Visual Solutions Pack добавлена возможность использовать модули VBX в приложениях dBASE for Windows. Вскоре после выхода системы программирования Delphi ожидается выход версии Visual Solutions Pack 1.2, включающей средства, ориентированные на Delphi.

▲ForeHelp

ForeHelp — система для быстрой разработки и построения справочных систем. в формате файлов справки Windows. **ForeHelp** лицензирована у фирмы ForeFront, Inc., Boulder CO.

В отличие от большинства других программ для построения справочных файлов Windows среда разработки ForeHelp работает в режиме WYSIWYG и не требует предварительного написания содержимого справочной системы с использованием текстового процессора.

С помощью ForeHelp можно создавать справочные подсистемы для задач, написанных на С++ и на языке dBASE для Windows.

▲Единство технологии

Границы семейства С++ очерчиваются достаточно условно, так как в целом программы компании базируются на единых технологических принципах, и, дополняя друг друга, продукты разных семейств часто используются совместно. Например, Visual Solutions Pack и ForeHelp вполне могут оказаться инструментами, используемыми программистом, работающим с dBASE for Windows или Delphi.

При создании большой системы у разработчика имеется возможность выбора продуктов, принадлежащих к различным семействам в качестве инструментов для изготовления компонент системы. Единый механизм работы с базами Borland Database Engine (BDE), входящий в комплект Borland C++ 4.5 and Database Tools, является общим для таких продуктов компании Borland, как Paradox 5.0 for Windows, dBASE 5.0 for Windows и Delphi. BDE и набор драйверов SQL Link, также входящий в Borland C++ 4.5 and Database Tools превращает эту систему программирования в средство построения приложений с архитектурой "клиент/сервер", дополняя "клиент/серверные" возможности dBASE и Paradox.

Поддержка в продуктах компании таких средств интеграции приложений, как работа с библиотеками динамической компоновки (DLL), технологии создания приложений с OLE 2.0 (Paradox, Delphi) и поддержки OLE-объектов (практически все продукты компании), VBX (dBASE, Delphi) позволяет компонентам системы, сделанным с помощью различных инструментов, непосредственно работать с общими объектами данных и совместно использовать общий программный код.

Таким образом, выделение перечисленных выше продуктов в отдельное семейство Borland C++ определяется в основном традициями. По сути, это семейство — неотъемлемая часть технологии Borland, одно из основных направлений в ее стратегии.

По материалам компании Borland

Как создать нестандартный элемент в библиотеке классов Turbo Vision

Дмитрий Рамодин

Канули в лету времена написания программ "с нуля". Незачем рисовать красивый интерфейс собственного сочинения, когда можно взять готовую библиотеку классов Turbo Vision и получить на выходе продукт "a la Borland". Нынешние библиотеки классов покрывают фактически все потребности программиста — от резидентных программ и инсталлятора до сложной трехмерной графики. К сожалению, поставщики уделяют мало внимания нетривиальным задачам, видимо, полагая, что их детище может все, да еще и пирожки печь. Недееспособность такой теории доказывает появление сопутствующей литературы, которая старается заполнить информационную брешь. Хорошим примером может служить книга А.Федорова "Borland Pascal: Практическое использование Turbo Vision 2.0". Она написана по принципу "Здесь то, что вы не смогли найти в стандартной документации". Кроме того, идеи этой книги прекрасно могут быть перенесены в C++.

Предположим, перед нами стоит задача написать небольшую программу с использованием библиотеки Turbo Vision 2.0, к примеру некий тест. Пусть наша программа выводит на экран диалоговую панель. Сверху на диалоговой панели должен быть показан вопрос, а под вопросом должен находиться список выбора с различными вариантами ответов. Это легко можно решить при помощи стандартного класса диалогового окна с вопросом вверху и списком выбора внизу. Только вот беда, элементы списка выбора могут быть больше одной строки, отличаться друг от друга количеством строк, да и вопрос может быть многострочным. Вот тут-то и пасует хваленая библиотека! Хотя, нет, ехидно улыбаясь, она говорит нам: "Что, как только задача чуть сложнее палки с веревкой, так сразу и руки вверх?" Да нет. Мы можем и подумать. Здесь уместен маленький совет: "Я настоятельно рекомендую Вам приобрести исходные тексты стандартной библиотеки компилятора Turbo Pascal. Этот продукт... включает в себя полный исходный текст библиотеки Turbo Vision 2.0. Ознакомление с реализацией библиотеки поможет Вам не только разобраться в том, как создаются коммерческие объектно-ориентированные библиотеки классов, но и послужит отличным источником знаний" (Федоров А. Borland Pascal: Практическое использование Turbo Vision 2.0, Киев, "Диалектика", 1993, с.3).

Первое, что необходимо для создания нестандартного класса, — это найти ему достойный класс-предок в иерархии. Для нашей задачи, пожалуй, уместен абстрактный класс TListViewer, который является шаблоном высокого уровня для всех списков выбора Turbo Vision 2.0. Следующий шаг — внимательно изучить поведение класса и определить, чего в нем не хватает, а что излишне. Умеет же класс TListViewer многое: он знает, какой элемент сейчас выделен курсором (переменная focused); знает число элементов списка (переменная range); умеет отображать самого себя (метод draw); обрабатывает сообщения клавиатуры и мыши (метод handleEvent) и т.д. Что и говорить, предок что надо! Однако он приспособлен отображать элементы размером только в одну строку, да и совершенно не знает, как обращаться с данными для списка, о чем говорит чистая функция getText. Первоначально полагалось, что данные будут храниться в виде коллекции, и метод getText будет научен извлекать их оттуда для метода draw, который будет выводить подставленный ему текст в списке на экране. Но поскольку мы делаем нечто нестандартное, то пусть данные хранятся в массиве указателей на связанные списки строк. Каждый такой список будет одним элементом выбора, и каждый элемент связанного списка строк будет размером в одну экранную строку. Значит, для вывода на экран такого элемента нужен совсем новый метод draw. Метод getText мы переопределим как пустую функцию. Это превратит абстрактный класс в обычный и не будет мешать нам. Связанный список удобно строить из объекта класса TSItem, который сам позаботится о выделении памяти под строку и об ее уничтожении. Замечательный метод TListViewer::selectItem пошлет сообщение владельцу списка (то есть диалоговой панели), в котором уведомит его о выборе пользователем некоего элемента списка выбора.

На листинге показан законченный вариант нашей задачи. Здесь мы объявляем два новых класса: TTextBlock, который будет отражать вопрос в диалоге и реализованный как потомок класса TView, и TBlockList, являющийся собственно списком выбора и основным объектом нашего внимания.

Кроме того, мы объявляем три глобальных функции: readFile, читающую определенное количество байт из определенного файла и изменяющую переменную-указатель в позиции в файле buildList, строя-

щую связанный список из элементов типа TSItem, и disposeList, этот список уничтожающую.

Данные для теста хранятся в трех файлах: answ.dat, quest.dat, choice.dat.

"answ.dat" хранит общее количество вопросов в тесте и номера правильных ответов. Формат его следующий: 2 байта — общее количество вопросов в тесте; N байт, количество которых равно числу, указанному в первых двух байтах файла, где каждый байт — номер правильного ответа минус 1.

"quest.dat" хранит тексты вопросов теста и имеет следующий формат: несколько элементов, где каждый элемент состоит из 1 байта-счетчика (указывает количество строк в элементе) и N строк текста, в количестве, указанном в байте-счетчике. Общее количество элементов равно числу в первых двух байтах файла "answ.dat".

"choice.dat" хранит элементы списка выбора и имеет следующий формат: несколько блоков, где каждый блок состоит из 1 байта-счетчика (указывает количество элементов в списке выбора); каждый элемент списка выбора состоит из 1 байта-счетчика (указывает количество строк в элементе) и N строк текста, в количестве, указанном в байте-счетчике. Общее количество блоков равно числу в первых двух байтах файла "answ.dat".

Следует отметить, что все файловые форматы взяты лишь для наглядности. В заключение хочу повторить: обязательно изучите исходные тесты вашей библиотеки! Только так вы сможете понять до конца идеологию и стратегию ее работы. Это касается любой библиотеки классов, будь то OWL или TurboProfessional. Успехов вам.

```
//----------------------
//
//   TEST.H
//
//----------------------


#ifndef __TEST_H
     #define __TEST_H

const int dWidth = 73;

const int cmNewTest = 200;   // Начать новый тест
const int cmPresent = 201;   // Найти окно на экране
const char *datFile = "choice.dat";
const char *qstFile = "quest.dat";
const char *ansFile = "answ.dat";

//---- Глобальные рабочие функции ------
long readFile(const char *, unsigned char *, long, long=2);
TSItem * buildList(long &, const char *);
void disposeList(TSItem *);

//---- Описания классов --------------
class TTextBlock : public TView
{
     short color;        // Цвет текста
     TSItem * question;  // Вопрос
     long questOff;      // Указатель позиции в файле
public:
     TTextBlock(const TRect&, short);
     ~TTextBlock();
     void reset(void);   // Начать снова
     void next(void);    // Подгрузить следующий вопрос
     virtual void draw(void);
};

class TBlockList : public TListViewer
{
     TSItem * list[5];   // Элементы списка выбора
     long  choiceOff;    // Указатель позиции в файле
public:
     TBlockList(const TRect&);
     ~TBlockList();
     void draw(void);
     void handleEvent(TEvent&);
     void reset(void);   // Начать снова
     void next(void);    // Подгрузить следующие элементы списка выбора
     void getText(char * dest, short item, short maxLen) {};
     short getFocused(void) {return focused;};
};

class TTestDlg : public TDialog
{
     short answer;          // Правильный ответ
     short right;              // Всего правильно отвечено на right вопросов
     short qCount;          // Сколько вопросов осталось
     short totCount;           // Всего вопросов в тесте
     TBlockList * selector;    // Наш список выбора
     TTextBlock * problem;     // Элемент диалога "Вопрос"
     long ansOff;              //  Указатель позиции в файле
public:
     TTestDlg::TTestDlg(const TRect&,const char*);
     virtual void handleEvent(TEvent&);
     void setUpDlg(void);     // Установить окно диалога
     void switchNext(void);   // Следующий вопрос
};



class TTestApp : public TApplication
{
     TTestDlg * win;           // Указатель на окно диалога
     void resetTest(void);     // Привести все в исходное состояние

public:
     TTestApp();
     static TStatusLine *initStatusLine( TRect r );
     static TMenuBar *initMenuBar( TRect r );
     virtual void handleEvent( TEvent& event);
     void newWindow(void);    // Смастерить диалог
     virtual void outOfMemory()
     {messageBox("\03Нет у меня больше\n\x03памяти!", mfError|mfOKButton);};
};

#endif //__TEST_H



//----------------------
//
//   TEST.CPP
//
//----------------------


#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>

#define  Uses_TEventQueue
#define  Uses_TEvent
#define  Uses_TProgram
#define  Uses_TApplication
#define  Uses_TKeys
#define  Uses_TRect
#define  Uses_TMenuBar
#define  Uses_TSubMenu
#define  Uses_TMenuItem
#define  Uses_TStatusLine
#define  Uses_TStatusItem
#define  Uses_TStatusDef
#define  Uses_TSItem
#define  Uses_TDeskTop
#define  Uses_TView
#define  Uses_TDialog
#define  Uses_TListViewer
#define  Uses_TStaticText
#define  Uses_MsgBox

#include <tv.h>
#include "test.h"
```

ЛААЛЬ

LAAL'E

NOVELL
Networking Partner

Системная интеграция

- Проектирование и поставка офисных, банковских и корпоративных сетей
- Создание локальных и распределенных сетей

EtherNet	FDDI
Token Ring	ATM
Fast EtherNet	X.25

√ Моделирование сетевых проектов в лаборатории

√ Поставка разветвленных систем хранения информации

√ "Горячая" техническая поддержка, консультации, обучение

√ Гарантийное обслуживание до 5 лет

√ Сертифицированные курсы:
- администрирование и сопровождение в сетях NOVELL
- теория и практика локальных сетей
- работа с Microsoft Office
- администрирование среды Lotus Notes

Оборудование фирм

Компьютеры: Compaq, Intel

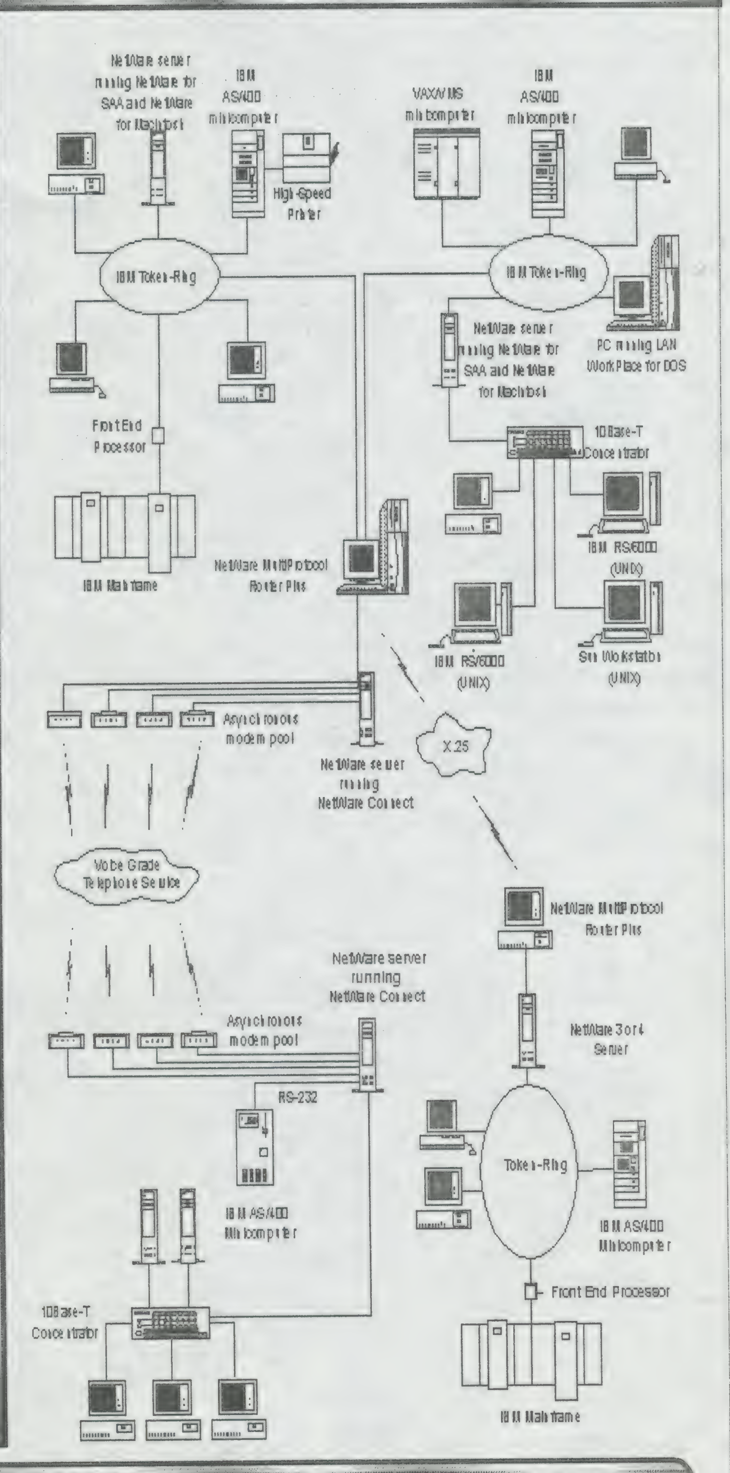
Notebooks: IBM, Toshiba

Сети: Novell, Microsoft, Artisoft, SynOptics, Wellfleet, AT&T, 3Com, D-Link, CNet

Периферия: Hewlett-Packard, Epson, Star, Logitech, APC, ZyXEL

Программное обеспечение: Borland, Lotus, Symantec, Corel, Autodesk, Aldus

Оргтехника: Rank Xerox



Тел.: (095) 273-56-71, 273-02-86; факс: (095) [illegible]8-14-60

```
//////////////////////////////
void TTestApp::handleEvent(TEvent&  event)
{
    TApplication::handleEvent(event);
    if( event.what == evCommand )
    {
        if(event.message.command==cmNewTest)  // Если новый тест
        {
            clearEvent(event);
            resetTest();
            newWindow();
        }
    }
}
//-Ищет старое окно и уничтожает его
void  TTestApp::resetTest(void)
{
    if(message(deskTop, evBroadcast, cmPresent, 0))
    destroy(win);
}

/////////////////////////////////
void  TTestApp::newWindow(void)
{
    win = new TTestDlg(TRect( 1, 1, 77, 22 ), "Ответьте на вопросы");
    win->setUpDlg();                  // Конструируем диалог
    deskTop->insert(validView(win));  // Показываем его
    win->switchNext();                // Подкачать первую порцию данных
}

//////////////////////////////
void  TTestDlg::handleEvent(TEvent&  event)
{
    TDialog::handleEvent(event);
    if(event.what == evBroadcast)
    {
        switch(event.message.command)
        {        // Нас ищут
            case cmPresent:
            clearEvent(event);
            break;                    // Пользователь выбрал пункт списка
            case  cmListItemSelected:
            {
                TBlockList * oops = (TBlockList*)event.message.infoPtr;
                if((oops->getFocused())!=answer)       //Узнаем, что выбрал
                                                       //пользователь
                {         // Если ответ неверен
                    right-=1;
                    messageBox("\x03Неправильный   ответ!\n\x03Продолжаем.",
                        mfError|mfOKButton);
                }
                qCount-=1;
                if(qCount) // Если есть еще вопросы
                {
                    switchNext();
                }
                else      // Если конец теста
                {
                    hide();  // Спрятать диалог
                    messageBox(mfInformation|mfOKButton,
                        "\x3Итого:\nПравильных  ответов
                        %d\nНеправильных ответов  %d", right,
                        totCount - right);
                    destroy(this); // Уничтожить диалог
                }
            }
        }
    }
}

/////////////////////////////////
void      TTestDlg::setUpDlg(void)
{
    unsigned char temp_buf[3];

    insert(new   TStaticText(TRect(2,2,9,3),"Вопрос:"));
    selector = new TBlockList(TRect(2,6,dWidth+1,20));
    insert(selector);
    problem = new TTextBlock(TRect(2,3,dWidth+1,6), 0x75);
    insert(problem);
    selector->reset();
    problem ->reset();
    ansOff = 0;
    ansOff = readFile(ansFile, temp_buf, ansOff,3);       // Узнаем количество
                                                          // вопросов
    temp_buf[3]='\0';
    right = qCount = totCount = atoi(temp_buf);
}
```

```
/////////////////////////////////
void  TTestDlg::switchNext(void)
{
    unsigned char temp_buf[2];
    TSItem * temp_link;
    ansOff = readFile(ansFile, temp_buf, ansOff);
    temp_buf[1]='\0';
    answer = atoi(temp_buf);
    problem ->next();
    selector->next();
    TProgram::deskTop->lock();
    redraw();           // Освежить изображение диалога
    TProgram::deskTop->unlock();
}
/////////////////////////////////
void  TTextBlock::reset(void)
{
    question = NULL;
    questOff = 0;
}

/////////////////////////////////
void  TTextBlock::next(void)
{
    if(question != NULL) disposeList(question); // Убить старый список
    question = buildList(questOff, qstFile);    // Сделать новый список
}

/////////////////////////////////
void  TTextBlock::draw(void)
{
    TDrawBuffer db;
    TSItem * temp;

    db.moveChar(0, ' ', color, dWidth);
    writeLine(0, 0, dWidth, size.y, db);
    temp = question;
    for(short i=0; i<size.y; i++)
    {
        if(!temp)break;
        db.moveStr(0, temp->value, color);
        temp = temp->next;
        writeLine(0, i, size.x, 1, db);
    }
}

/////////////////////////////////
void  TBlockList::reset(void)
{
    choiceOff = 0;
    for(int i=0; i<5; i++) list[i] = NULL;
}

/////////////////////////////////
void  TBlockList::next(void)
{
    unsigned char temp_buf[2];

    focused=0;
    for(int i=0; i<range; i++)
    {
        if(list[i] != NULL)
        {
            disposeList(list[i]);             // Убить старые списки
            list[i] = NULL;
        }
    }
    choiceOff = readFile(datFile, temp_buf, choiceOff);
    range = atoi(temp_buf);
    for(i=0; i<range; i++) list[i] = buildList(choiceOff, datFile);
                        // Построить новые списки элементов выбора
}

/////////////////////////////////
void  TBlockList::draw(void)
{
    int cur_line = 0;
    ushort n_col = getColor(1);               // Нормальный цвет элемента
    ushort f_col = getColor(3);               // Подсвеченный цвет элемента
    TDrawBuffer db;
    TSItem * temp;

    db.moveChar(0, ' ', n_col, dWidth);
    writeLine(0, 0, dWidth, size.y, db); // Нарисовать фон списка выбора
    for(short i=0; i<range; i++)
    {
        temp = list[i];
```

```
        while(temp)
        {
            db.moveStr(0,  temp->value,  isSelected(i)?f_col:n_col);
            temp = temp->next;
            writeLine(0, cur_line++, size.x, 1, db);
                                    // Рисуем элементы списка выбора
        }
    }
}

//////////////////////////////////////////
void TBlockList::handleEvent(TEvent & event)
{
    if(((event.what==evKeyDown)&&(event.keyDown.keyCode==kbEnter))||
            ((event.what==evMouseDown)&&(event.mouse.doubleClick)))
    {
                        // Если пользователь выбрал ответ,
                        // то уведомить владельца списка выбора
        selectItem(focused);
        clearEvent(event);
    }
    else
    {
        TListViewer::handleEvent(event);
        TProgram::deskTop->lock();
        drawView();
        TProgram::deskTop->unlock();
    }
}

//////////////////////////////////
TTestApp::TTestApp() :
        TProgInit(  &TTestApp::initStatusLine,
            &TTestApp::initMenuBar,
            &TTestApp::initDeskTop),
        win(NULL)
{  messageBox("\03Написано\n\n\x03Рамодиным Дмитрием\n\n\x03  1995г.  ",
        mfInformation|mfOKButton);   }

////////////////////////////////
TTextBlock::TTextBlock(const TRect& bounds, short aColor):
          TView(bounds), color(aColor) {}

////////////////////////////////
TTextBlock::~TTextBlock()
{
    if(question != NULL) disposeList(question); // Явно уничтожаем связанные
                                                // списки
}

////////////////////////////////
TBlockList::TBlockList(const TRect& bounds)
        TListViewer(bounds, 1, NULL, NULL) {}

////////////////////////////////
TBlockList::~TBlockList()
{
    for(int i=0; i<range; i++)
    {
        if(list[i] != NULL) disposeList(list[i]);       // Явно уничтожаем
                                                        // связанные списки
    }
}

////////////////////////////////
TTestDlg::TTestDlg(const TRect& bounds,const char* aTitle):
    TDialog(bounds, aTitle),  TWindowInit(&TTestDlg::initFrame)
{ flags &= ~(wfClose|wfMove); }                  // Не закрывать и не двигать


//////////////////////////////// Debugged ////////////////////
TMenuBar *TTestApp::initMenuBar( TRect r )
{
    r.b.y = r.a.y + 1;    // set bottom line 1 line below top line
    return new TMenuBar( r,
        *new TSubMenu( "~Т~ест", 0 )+
        *new TMenuItem( "~Н~овый", cmNewTest,  kbF3, hcNoContext, "F3" )+
        newLine()+
        *new TMenuItem( "~В~ыход", cmQuit,  kbAltX, hcNoContext, "Alt-X" ));
}

///////////////////////////
TStatusLine *TTestApp::initStatusLine( TRect r )
{
    r.a.y = r.b.y - 1;
    return new TStatusLine( r,
        *new TStatusDef( 0, 0xFFFF ) +
        *new TStatusItem( 0, kbF10, cmMenu ) +
        *new TStatusItem( "~Alt-X~ Выход", kbAltX, cmQuit ));
}

TSItem * buildList(long & positionVar, const char * file)
{
    unsigned char temp_buf[dWidth+1];
    int count;
    TSItem * temp_head, * temp_tail;
    positionVar = readFile(file, temp_buf, positionVar); // Считать
                                        //количество строк в списке
    count = atoi(temp_buf);
    positionVar = readFile(file, temp_buf, positionVar, dWidth);
    temp_head = temp_tail = new TSItem ((char *)&temp_buf[0],NULL);
    for(int i =1; i<count; i++)
    {
        positionVar = readFile(file, temp_buf, positionVar, dWidth);
        temp_tail->next = new TSItem ((char *)&temp_buf[0],NULL);
        temp_tail = temp_tail->next;
    }
    return temp_head;
}

/////////////////////////////
void disposeList(TSItem * aList)
{
    TSItem * temp_head, * temp_tail;
                        // Если новый тест
    for(temp_head = temp_tail = aList;
    temp_head->next != NULL; )
    {
        temp_tail = temp_head->next;
        delete temp_head;
        temp_head = temp_tail;
    }
    delete temp_head;
}

/////////////////////////////
long readFile(const char * file, unsigned char * buffer,
        long offset, long number)
{
    static ifstream ifs;
    long temp;
    ifs.open(file)                  // Открыть поток
    ifs.seekg(offset, ios::beg);    // Установить позицию указателя файла
    ifs.get(buffer, number);        // Считать данные из потока
    temp=ifs.tellg();               // Считать позицию указателя файла
    ifs.close();                    // Закрыть поток
    return temp;                    // Вернуть позицию указателя файла
}

//////////////////////
int main()
{
    TTestApp myApp;
    myApp.run();
    return 0;
}
```

Paradox 5.0 для Windows. Новые возможности

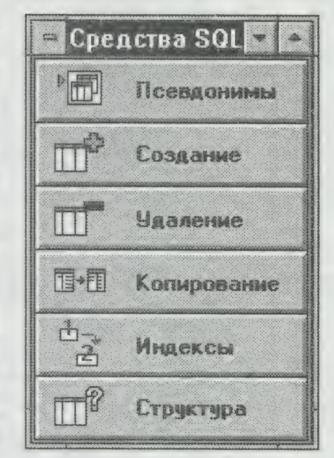
Сергей Каратыгин, Аркадий Тихонов

В августе 1994 года компания Borland Int. выпустила новую версию системы управления базами данных (СУБД) Paradox 5.0 for Windows. На протяжении долгих лет мы уже привыкли, что Paradox олицетворяет последние достижения в системах управления базами данных. Каждая новая версия становится все более быстрым и мощным инструментом, предоставляя дополнительные средства как конечным пользователям, так и разработчикам приложений. Paradox 5.0 for Windows продолжает традиции, сочетая в себе простоту использования с уникальными средствами для разработки законченных приложений. К моменту выхода этой статьи уже должна быть объявлена локализованная версия Paradox 5.0 для Windows.

Давайте попытаемся оценить новые возможности, предоставляемые Paradox 5.0, хотя следует сразу отметить, что сделать это в рамках одной статьи представляется задачей практически невыполнимой.

▲Расширение возможностей

- В Paradox 5.0 значительно повышены быстродействие и производительность системы. Уменьшено время загрузки программ, а также увеличены скорость компиляции и исполнения приложений на ObjectPAL.
- Благодаря новой запатентованной технологии SpeedFilters увеличена скорость выполнения запросов по образцу (QBE).
- Встроенная в Paradox технология IDAPTER позволяет Paradox 5.0 использовать для доступа к данным любой драйвер стандарта ODBC.
- Для связи с удаленными базами данных на SQL-серверах Paradox использует высокопроизводительные драйверы SQL Links (в версии 5.0 поставляются отдельно), которые реализуют все возможности серверов баз данных, напрямую обращаясь к API-интерфейсу сервера. Такой подход позволяет достичь максимальной производительности при работе с базами данных InterBase, Informix, SYBASE, Oracle и MS SQL Server.
- Для взаимодействия с удаленными таблицами может быть использован специальный инструмент — “Средства SQL”. Теперь таблицы и индексы на сервере базы данных могут быть созданы непосредственно из Paradox.



- В Paradox 5.0 имеется встроенная технология обмена объектами Borland Object Exchange, предназначенная для обмена информацией через системы электронной почты и локальные сети. Поддерживается широкий спектр протоколов транспортного уровня для передачи сообщений: MAPI, MHS, WPO, cc:Mail и VIM.
- Требования к аппаратным средствам остались прежними, за исключением требований к минимальному объему оперативной памяти. Теперь для работы требуется не менее 6 Мбайт, а для выполнения приложений рекомендуется 8 Мбайт оперативной памяти.

▲Новые типы полей

В Paradox 5.0 введены новые типы полей:
- счетчик;
- дата и время;
- время;
- логический;
- битовый;
- длинное целое.

Тип
A Текстовый
N Числовой
$ Денежный
S Целое
I Длинное целое
Фиксированный
D Дата
T Время
@ Дата и время
M MEMO
F Форматируемый
G Графический
O Объект OLE
L Логический
+ Счетчик
B Двоичный
Y Байтовый

Тип поля “Счетчик” удобно использовать в справочных таблицах в качестве идентификатора записи (например, кода товара). Данный тип поля избавит вас от необходимости изменения кода товара при каждом добавлении в

справочник нового товара. Тип данных “Дата и время” обеспечивает совместимость с серверами баз данных. Типы данных “Логический” и “Битовый” удобны для хранения и дальнейшей обработки логических данных и данных, состоящих из одного символа. Поля типа “Длинное целое” позволяют вам хранить в числе до 32 десятичных значащих цифр.

Средства для начинающих пользователей

В Paradox 5.0 большое внимание уделено обучению начинающих пользователей работе с системой. Для этих целей введены новые средства: инструкторы и эксперты.

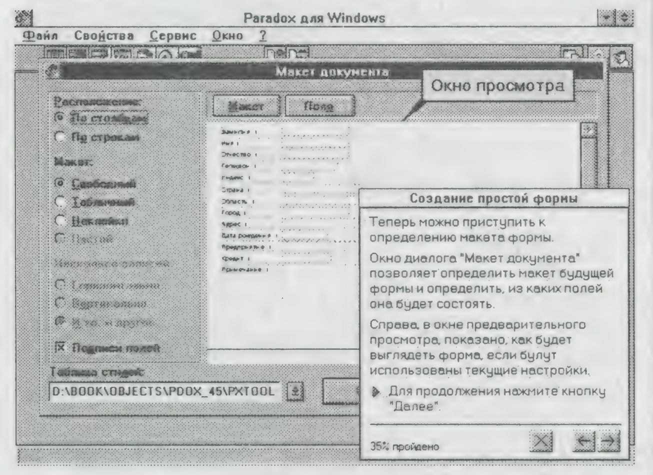


Рис. 1. Урок по созданию простой формы

Инструкторы обучают начинающего пользователя основным операциям, таким как создание таблиц, форм и отчетов. Вызов инструктора осуществляется нажатием кнопки **Инструктор** на панели управления. Инструктор предлагает на ваш выбор четыре основных раздела обучения:

- введение в Paradox;
- создание базы данных;
- работа с таблицами;
- запросы, формы и отчеты.

Каждый из разделов содержит собственные темы. Инструктор не просто обучает работе с Paradox, но и помогает создавать необходимые таблицы, формы, отчеты и многое другое. Повторив несколько раз вместе с инструктором операции создания таблиц, форм, отчетов, вы будете в состоянии в дальнейшем проделать аналогичные операции самостоятельно.

Во время урока инструктор отображает на экране краткие пояснения и перечень операций, которые необходимо выполнить на данном шаге (рис. 1). Чтобы облегчить пользователю ориентацию в рабочей области, на экране отображаются указатели объектов, на которые ссылается инструктор. При ошибочных действиях инструктор открывает дополнительное окно, в котором настойчиво предлагает выполнить указанное действие.

Эксперты могут пригодиться как начинающим пользователям, так и пользователям, уже имеющим навык работы с Paradox. Эксперт проделает за вас все необходимые шаги по подготовке конструкторских документов (форм и отчетов). При этом для создания почтовых этикеток предусмотрен собственный эксперт. В процессе создания документа эксперты предлагают на ваш выбор ряд шаблонов (рис. 2). При наличии большого выбора шаблонов создание отчета или формы требуемого формата не составит большого труда.

Для вызова эксперта достаточно нажать кнопку **Эксперт** на панели управления и выбрать требуемый эксперт из панели управления экспертной системы. Далее вместе с экспертом вы пройдете все шаги по формированию конкретного документа. Например, при создании формы эксперт поможет вам выбрать, в каком виде будут располагаться данные в форме, а затем определить перечень таблиц, входящих в форму, и список отображаемых полей. Далее вы сможете выбрать элементы оформления полей формы. После завершения создания формы вы можете запустить ее на выполнение или перейти в режим конструктора для дальнейшей работы над формой. В дальнейшем созданная с помощью эксперта форма может дополняться программами обработки событий и объектами интерфейса.

Все эксперты Paradox 5.0 реализованы на языке ObjectPAL, а информация об экспертах содержится

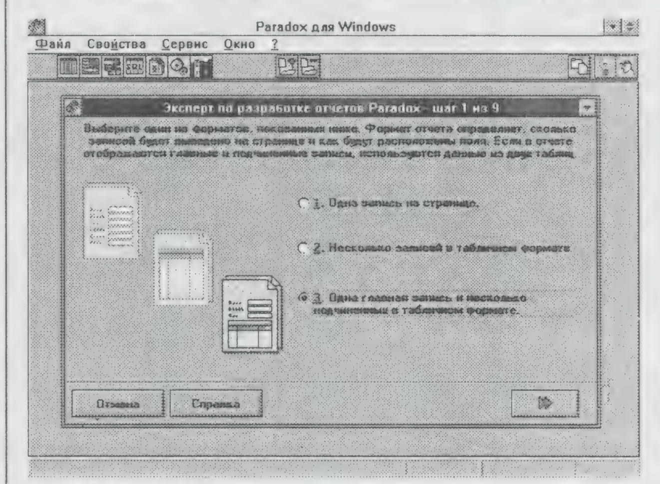


Рис. 2. Начальная стадия создания отчета с помощью эксперта

в таблицах, хранящихся в каталоге EXPERTS. Такой подход позволяет вам добавить в уже существующие эксперты новые шаблоны и даже создать собствен-

ных экспертов, которые будут доступны другим пользователям.

▲Графическая модель данных

В процессе создания форм и отчетов вам необходимо создавать модели данных. В Paradox 5.0 модель данных является отдельным объектом. Вы можете создать модель данных независимо от форм и отчетов, а затем использовать ее не только при создании конструкторских документов, но и запросов. Поскольку связи между таблицами в конкретном приложении обычно не изменяются, такой подход сэкономит вам значительное время при разработке форм, отчетов и запросов.

Создание модели данных осуществляется в окне диалога "Шаблон схемы данных", которое открывается в результате выбора **Сервис | Схема данных**. На первом этапе определяются таблицы, входящие в модель данных. Далее между таблицами устанавливаются связи. Для этого вам достаточно протянуть нить между связываемыми таблицами и определить совпадающие поля. Если в указанных таблицах имеются поля с одинаковыми наименованиями и типами полей, связь установится автоматически. В противном случае вам придется установить связь самостоятельно.

Все действия по определению модели данных выполняются с помощью графического интерфейса. При такой организации диалога с этой нетривиальной задачей сможет справиться даже начинающий пользователь, а разработчик приложений сэкономит время для решения других проблем.

Файл, содержащий модель данных, может использоваться при создании сложных форм, отчетов, запросов и даже новых моделей данных. При использовании ранее созданной модели данных вместо выбора таблиц и создания модели данных отдельно для формы, отчета или запроса вы просто выбираете нужную модель из списка. При этом запрос, созданный с использованием модели данных, содержит не только таблицы, входящие в модель, но и связующие элементы.

▲Панель проектов

Панель проектов является дальнейшим развитием механизма папок, который использовался в предыдущих версиях Paradox для Windows, и позволяет организовывать все объекты приложения (формы, таблицы, отчеты и т. п.) в виде упорядоченных структур с расположением информации о каждом виде объектов в отдельном подразделе. Благодаря такому подходу ускоряется выбор требуемого объекта и отображаются только те объекты, которые размещены в папке. Кроме непосредственного присутствия объекта в папке в нее могут быть помещены объекты, связанные с объектами папки по ссылке. Данный подход позволяет многократно использовать одни и те же объекты в разных приложениях.

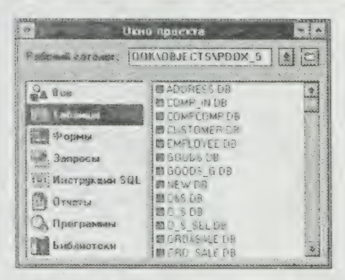


Рис. 3. Панель проектов

Панель проектов (рис. 3) содержит окно с пиктограммами основных объектов Paradox и информационное окно со списком объектов выбранного типа. При работе с панелью проектов с помощью правой кнопки мыши вы можете вывести на экран меню возможных действий с объектом определенного типа и выбрать требуемое действие, например, (для формы) просмотреть данные, перейти в режим конструктора и т. п.

▲Новый отладчик ObjectPAL

Если инструкторы и эксперты полезны прежде всего начинающим пользователям, то новый отладчик ObjectPAL предоставляет расширенные средства для опытных разработчиков. Отладчик позволяет устанавливать контрольные точки, создавать окна слежения за изменением значений переменных, осуществлять трассировку приложений и т. д. Теперь можно сказать, что в Paradox появился отладчик, который по своей мощности и организации близок к отладчикам, используемым в системах проектирования типа C++.

Для установки контрольных точек достаточно подвести курсор к требуемой строке и выбрать кнопку

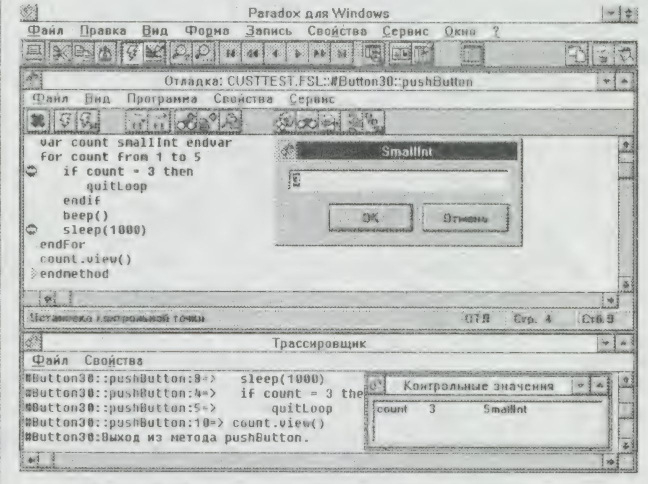


Рис. 4. Окна отладчика, трассировщика, просмотра значений и слежения за переменными

установки контрольной точки. Строка, содержащая контрольную точку, имеет слева метку в виде специального знака. С помощью окна "Контрольные точки" вы можете просмотреть все установленные в данный момент контрольные точки с указанием их местоположения.

В процессе выполнения формы при достижении контрольной точки выполнение программы приостанавливается и открывается окно отладчика (рис. 4). В окне отладчика вам доступны следующие средства:

- остановка выполнения;
- переход в режим просмотра;
- выполнение программы до конца текущего метода;
- выполнение очередного шага без захода в процедуры и методы;
- выполнение очередного шага с заходом в процедуры и методы;
- добавление переменной в окно слежения за переменными;
- установка/снятие контрольной точки;
- просмотр значения переменной;
- открытие/закрытие окна отладчика;
- открытие/закрытие окна слежения за переменными;
- открытие/закрытие окна контрольных точек;
- открытие/закрытие окна трассировки;
- открытие/закрытие окна стека вызова.

▲Технология Live Query View

Запрос в Paradox 5.0 может выполняться стандартным образом или с использованием технологии Live Query View. Это средство доступно при выполнении запросов как к локальным, так и удаленным данным. Что же такое Live Query View?

При выполнении запроса Paradox помещает результат выборки в таблицу ANSWER. Связь между исходной таблицей и таблицей ANSWER после выполнения запроса не поддерживается. Если вы начнете редактировать таблицу ANSWER, это никак не отразится на содержимом исходной таблицы.

При выполнении запроса с использованием Live Query View результат выборки помещается в динамический набор, который поддерживает связь с исходной таблицей. Если теперь вы начнете редактировать данные в результирующем наборе, сделанные вами изменения автоматически будут перенесены в исходную таблицу. При этом вы сразу сможете наблюдать результаты редактирования в исходной таблице. Поскольку в результате запроса выбираются только записи, удовлетворяющие определенным критериям, технология Live Query View предоставляет в ваше распоряжение эффективное средство редактирования ограниченных наборов записей.

Для редактирования исходной таблицы с использованием результата выборки необходимо установить свойство **Динамический набор** в окне диалога "Результат выполнения запроса". Если результат выборки невозможно использовать для редактирования исходной таблицы, динамический набор открывается только для чтения.

Кроме Live Query View в Paradox 5.0 использована новая технология фильтров, обеспечивающая доступ только к тем данным, которые удовлетворяют заданным критериям. Теперь фильтр может быть установлен не только во время просмотра таблицы, но также и при создании формы.

▲OLE в Paradox 5.0

Предыдущие версии Paradox для Windows позволяли осуществлять связь с внешними приложениями с использованием динамического обмена данными (Dynamic Data Exchange — DDE) и механизма работы с составными документами (Object Linking and Embedding - OLE). При этом Paradox мог выступать в роли как сервера, так и клиента при динамическом обмене данными, а также OLE-клиента.

В новую версию включена полная поддержка стандарта OLE 2.0. Paradox может теперь выступать в роли не только OLE-клиента, но и в роли OLE-сервера. Данное средство позволяет вам включать таблицы Paradox в документы текстовых процессоров (например, Microsoft Word) и редактировать их, не покидая текстового процессора.

Действия, которые необходимо выполнить при установке связи в приложении, являющемся OLE-клиентом, зависят от конкретного приложения. Например, в Microsoft Word 6.0 вам необходимо открыть исходный документ и выбрать команду **Вставка | Объект**. Далее в окне диалога "Объект" следует перейти на страницу "Создать новый" и указать тип объекта — Paradox Table.

После автоматической загрузки Paradox укажите таблицу, данные которой необходимо перенести на OLE-сервер. Для возврата в Microsoft Word закройте выбранную таблицу. Теперь таблица Paradox включена в документ Microsoft Word. Для таблицы редактирования вам достаточно два раза нажать мышью ее изображение в документе. В результате управление будет передано Paradox, в котором автоматически открывается исходная таблица. Если к этому моменту Paradox не был загружен, он загрузится в фоновом режиме. После этого вы можете редактировать содержимое таблицы.

В связи с тем, что Paradox 5.0 может выступать в роли OLE-сервера, в ObjectPAL добавлены новые методы и процедуры для объекта типа OLE.

▲ Использование SQL в приложениях

Новым и очень мощным средством Paradox 5.0 является возможность выполнения SQL-выражений не только при работе с удаленными данными, но и с локальными таблицами. Такой подход еще более упрощает перенос приложений на архитектуру клиент/сервер и делает Paradox привлекательным для разработчиков, имеющих опыт работы с SQL-выражениями. Конструкция SQL более читабельна, чем конструкция Query, и, кроме того, является стандартным языком управления данными СУБД.

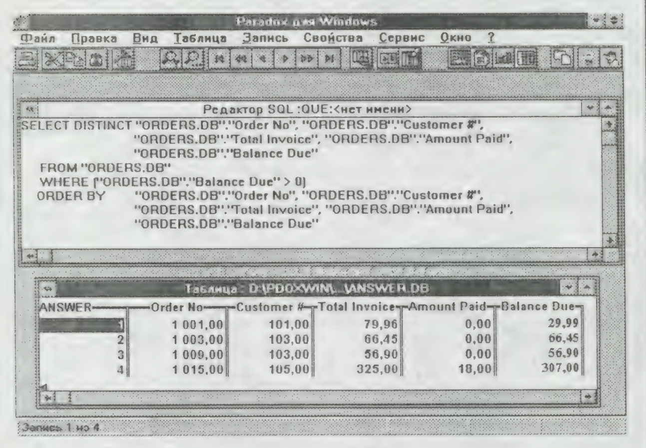


Рис. 5. Выполнение SQL-выражения

Paradox предоставляет вам возможность выполнения SQL-запросов в интерактивном режиме. Для этого выберите кнопку **SQL** на панели управления и введите в окне "Редактор SQL" SQL-выражение (рис. 5). Для выполнения запроса нажмите кнопку **Запуск SQL** на панели управления. Результат выполнения будет сохранен во временной таблице ANSWER, как и при выполнении обычных запросов по образцу. Созданное SQL-выражение может быть сохранено в файле для дальнейшего использования.

Если вы не знакомы с SQL-выражениями, но имеете навык формирования запросов по образцу, Paradox 5.0 предоставляет в ваше распоряжение средство для создания SQL-выражения на основании запроса. С его помощью вы сможете быстро и без ошибок создать SQL-выражение, не имея ни малейшего представления о его синтаксисе. Для этого откройте ранее созданный запрос или создайте новый, проверьте его, а затем выполните команду **Запрос | Инструкции SQL**. В результате будет открыто новое окно редактора SQL, содержащее текст SQL-выражения.

▲ Справочная система Paradox

Справочная система Paradox также претерпела изменения по сравнению с предыдущими версиями. Прежде всего изменилось отображение выбранных тем (рис. 6). Теперь для каждого выбранного метода существуют три контекстные ссылки: "См. также", "Пример" и "Тип объекта", к которому относится метод. При выборе "Пример" открывается новое окно, содержащее пример, соответствующий выбранной теме. Вы можете скопировать его во временный буфер Windows и использовать при разработке приложения.

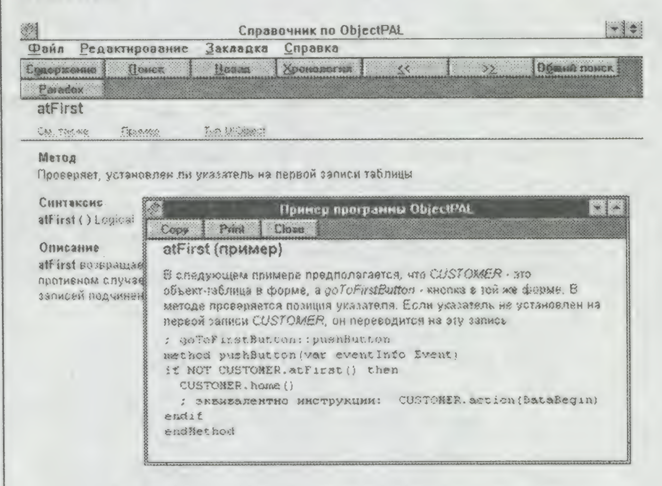


Рис. 6. Отображение выбранной темы

В справочной системе новой версии появилась возможность распечатывать описание выбранных методов. Вы можете выбирать методы в алфавитном порядке или по типам объектов. Для выбранного варианта можно указать все методы или выбрать только часть из них.

▲ Вместо заключения

В Paradox 5.0 появился ряд новых методов, а часть уже знакомых вам методов претерпела изменения. Более 100 новых методов ObjectPAL и 200 новых свойств объектов — этим грандиозным фактом мы закончим наш краткий обзор Paradox 5.0 для Windows. Таким образом, вместо заключения у нас получилось приглашение к новому обсуждению.

В 1995 году издательством "Бином" запланирован выпуск двух книг:

1. Каратыгин С.А. и Тихонов А.Ф. Работа в Paradox 5.0 для Windows на примерах. Локализованная версия. — II квартал.
2. Разработка приложений в Paradox 5.0 для Windows / Перевод с английского (Название приведено условно). — III квартал.

Мы надеемся, что эти книги будут полезны как конечным пользователям, так и разработчикам законченных приложений.

dBASE для Windows: объектно-ориентированные средства разработки

Егор Яковлев

В последние годы объектно-ориентированная технология разработки программ революционизировала способы и методы создания комплексных приложений. Уже нет необходимости доказывать, что использование этой технологии качественно повышает производительность программистов, в то же время значительно увеличивая надежность разрабатываемых программ. Использование стандартных библиотек объектов с заданными свойствами, реализующих типичные операции с данными, значительно сокращает объем кодирования и позволяет повторно использовать код уже отлаженных объектов. До недавнего времени все эти возможности были доступны лишь программистам, использующим объектно-ориентированные языки третьего поколения, такие как C++ или Pascal with Objects.

С выходом в августе 1994 года Windows-версии dBASE 5.0 разработчики приложений в области баз данных получили действительно новый объектно-ориентированный язык программирования, являющийся в то же время полностью совместимым со стандартом языка xBASE DOS-версий продукта.

Проект по реализации нового языка dBASE разработчики компании Borland начали со стандарта dBASE for DOS. Тем самым была обеспечена 100% совместимость на уровне уже разработанных программ. Далее, в язык были добавлены расширения, полностью реализующие все возможности интерфейса Windows и позволяющие напрямую обращаться к вызовам Windows API. Для полной поддержки функций Windows в языке реализована управляемая событиями модель программирования. В язык добавлен полный набор объектно-ориентированных расширений. Новый стандарт языка СУБД dBASE для Windows — единственный специализированный язык для разработки в области баз данных, в котором реализованы три основных постулата объектно-ориентированного программирования: инкапсуляция, наследование и полиморфизм. В результате создана среда программирования, сочетающая легкость использования с полной реализацией объектно-ориентированного подхода к разработке приложений. Кроме того, к моменту публикации этой статьи пользователям уже будет доступна полностью локализованная версия продукта.

Какие конкретно особенности позволяют говорить о dBASE как о полностью объектно-ориентированной среде разработки? Кратко перечислим средства dBASE, реализующие объектно-ориентированный подход к разработке приложений.

- Классы объектов, позволяющие создавать множественные экземпляры (instances) объектов, каждый из которых наследует те же самые характеристики.
- Определяемые пользователем классы объектов, позволяющие определять объекты с уникальными свойствами, первоначально отсутствовавшие в системе.
- Инкапсуляция, позволяющая связывать данные и методы в рамках одного объекта и реализующая защиту данных объекта от внешнего воздействия.
- Подклассы объектов, разделяющие часть или все данные и методы одного или нескольких классов-предков. Процесс порождения новых классов на основе уже определенных в системе называется *наследованием*.
- Полиморфизм — свойство, позволяющее передавать одно и то же сообщение разным объектам, и даже разным классам объектов. Объекты, получившие сообщение, обрабатывают его в зависимости от своих индивидуальных особенностей.

Далее на простых примерах рассмотрим конкретную реализацию этих средств в языке dBASE для Windows.

▲Инкапсуляция

dBASE для Windows позволяет запустить несколько экземпляров одной формы одновременно. В примере каждый экземпляр формы CLICK.WFM является реализацией одного и того же объекта, инкапсулирующего в себе и программу, и данные. На рис. 1 вы видите несколько экземпляров формы, каждый из которых содержит свою собственную реализацию данных. Число на кнопке означает количество нажатий на эту кнопку. Это простой пример инкапсуляции методов и данных для каждого экземпляра объекта-формы.



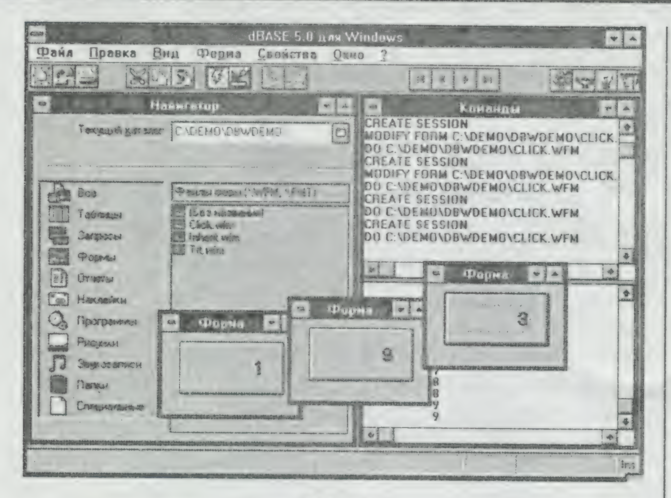


Рис. 1

▲Наследование

dBASE для Windows позволяет определять новые объекты на основе уже заданных, при этом наследуя все свойства и особенности объекта и добавляя новые. В примере программы показано, как на основе формы-предка INHERIT.WFM задается новая форма, имеющая дополнительный объект — кнопку, при этом наследуются все объекты и методы исходной формы.

```
set procedure to inherit.wfm
x1=new inherit()
    x1.text="Форма  просмотра"
    x1.text2.text="Табличный  просмотр"
    x1.top=2
    x1.left=2
    DEFINE PUSHBUTTON BUTTON2 OF x1;
        PROPERTY;
        Text "Табличный просмотр",;
        Width 14.00,;
        Height 2.00,;
        Top 12.00,;
        Left 13.00
x1.open()
WAIT "Нажмите любую клавишу..." TO dummy
x1.close()
```

▲Полиморфизм

Для начала определим два класса с именами "Павел" и "Кирилл" на основе предопределенного в dBASE для Windows класса FORM.

```
*   POLY2.PRG
CLASS ПАВЕЛ OF FORM
    this.text="Паша  Красюк"
    FUNCTION Привет
        this.text="Привет  от  Паши!"
ENDCLASS

CLASS КИРИЛЛ OF FORM
    this.text="Кирилл  Раннев"
    FUNCTION Привет
        this.text="Добрый  день!"
ENDCLASS
```

Далее в окне команд dBASE создадим два объекта каждого из классов.

```
SET PROCEDURE TO POLY2.PRG
Паша     = new Павел()
Кирилл   = new Кирилл()
Паша.Open()
Кирилл.Open()
```

Мы создали два экземпляра объектов разных классов, при этом каждый из них имеет собственную реализацию метода "Привет". Продемонстрируем это, набрав в окне команд dBASE следующую команду:

```
Павел.Привет()
```

При этом заголовок одной из форм изменился на "Привет от Паши!". Набрав ту же самую команду для объекта "Кирилл", **Кирилл.Привет()**, мы видим, что форма — "Кирилл" отвечает: "Добрый день!" (рис. 2).

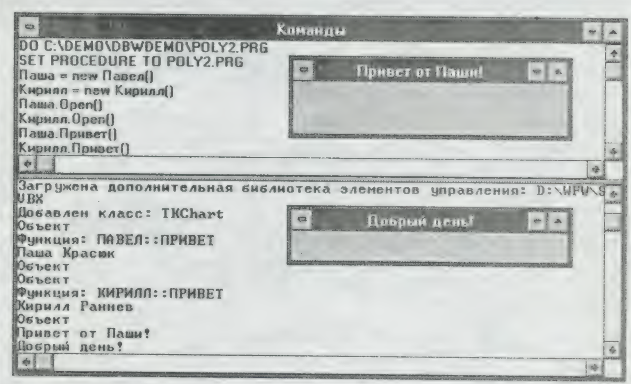


Рис. 2

Этот простой пример демонстрирует реальные возможности по использованию полиморфизма при разработке приложений в dBASE для Windows. Представим себе, что у нас имеется программа, реализующая систему учета кадров компании. Далее, у нас имеются постоянные работники, служащие, нанятые по контракту, а также приходящие рабочие, получающие почасовую оплату. Было бы действительно удобно начислить зарплату всем этим категориям работников, набрав, например, одну команду: "Зарплата". И это действительно возможно в dBASE. В приведенном примере каждый из объектов может произвести расчет заработной платы в соответствии со своими уникальными характеристиками. Такой подход значительно упрощает программирование, позволяя единым образом работать с объектами различной природы.

Далее рассмотрим некоторые другие мощные инструментальные средства для разработки программ в dBASE для Windows.

▲Препроцессор

Реализованный в dBASE препроцессор по большей части напоминает препроцессор языка С. В момент компиляции программы препроцессор просматривает код в поисках специальных директив и исполняет

их. Препроцессорные директивы позволяют включать в текст внешние файлы, определять константы, вычисляемые выражения и проводить условную компиляцию программы.

▲Отладчик

Одним из наиболее мощных и удобных в использовании средств является отладчик программ dBASE для Windows. Компанией Borland реализован прак-

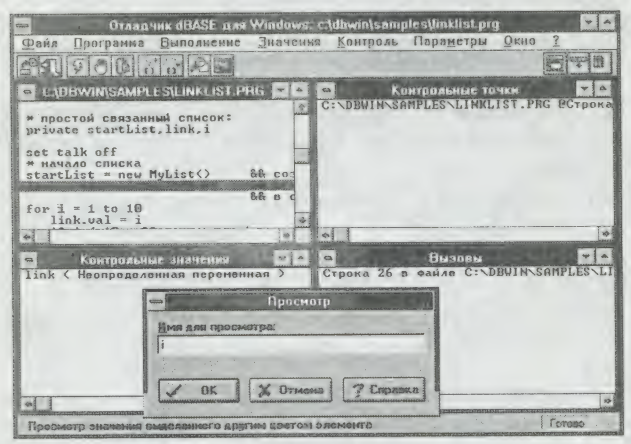


Рис. 3

тически полный набор стандартных функций для отладки программ на языке dBASE, включая точки останова, окна для просмотра значений переменных в процессе выполнения программы, стек вызовов процедур и функций и другие возможности (рис. 3).

▲Внутренние массивы

СУБД dBASE для Windows позволяет программисту определять массивы любой размерности, при этом для всех элементов массива могут выполняться операции поиска, сортировки, или удаления. Массивы dBASE могут интерпретироваться как объекты для доступа к методам и свойствам. Другая важная для реализации многих задач возможность — поддержка в dBASE разреженных (sparse) массивов данных. Такие массивы позволяют, например, хранить данные о пространственных измерениях, при этом поддерживая изменяемое количество непрерывных параметров распределения.

▲Передача параметров

dBASE поддерживает расширенные возможности передачи параметров в процедуры и функции. Можно передавать до 255 параметров, массивы и объекты, использовать при вызове количество параметров, отличное от заданного в определении процедуры, а также передавать параметры как по ссылке, так и по значению.

▲Специальные режимы тестирования

Новое мощное средство dBASE Coverage Analysis позволяет разработчикам при выполнении программы точно определять, какая часть кода программы исполняется. При этом генерируется сводный отчет. Эта новая возможность особенно полезна при тестировании законченного приложения, так как позволяет производить полное тестирование всего кода программы с регистрацией процесса исполнения.

▲Автоматическая генерация тестовых баз данных

Новая команда dBASE — GENERATE — позволяет в автоматическом режиме генерировать тестовые базы данных. Для этого достаточно открыть любую таблицу в режиме редактирования и далее выполнить команду с числовым аргументом, определяющим, сколько записей должно быть добавлено в таблицу. При этом dBASE строит добавляемые записи случайным образом. Например, команда GENERATE 100 заполнит все поля текущей таблицы (за исключением MEMO, OLE и двоичных полей) случайно построенными данными для новых ста записей таблицы.

▲Borland Database Engine

И dBASE для Windows, и СУБД Paradox 5.0 для Windows используют для доступа к данным общее ядро процессора баз данных — Borland Database Engine. Это означает, что работая с dBASE, пользователи могут открывать, создавать новые таблицы, исполнять запросы и модифицировать данные в таблицах формата Paradox. Более того, при работе в сети пользователи dBASE и Paradox могут одновременно работать с одними и теми же данными на файл-сервере. С использованием драйверов SQL Link, уже известных пользователям Paradox для Windows, пользователи dBASE получают доступ к данным на SQL-серверах, таких как Borland InterBase, Oracle, Informix, Sybase и Microsoft SQL Server. Такой подход позволяет разрабатывать приложения в архитектуре "клиент/сервер" с использованием стандартных средств dBASE для Windows (рис. 4).

▲Ссылочная целостность

dBASE для Windows поддерживает ссылочную целостность данных через новую реализацию команды SET



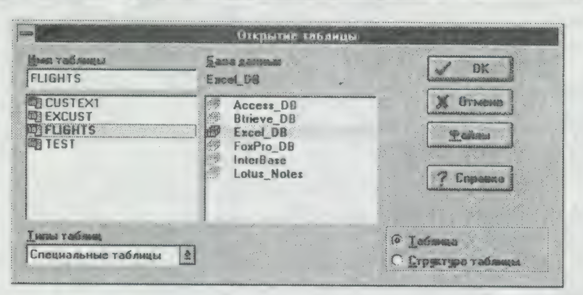


Рис. 4

RELATION, которая определяет отношения между таблицами dBASE. Опция CONSTRAIN ограничивает доступ к данным подчиненной таблицы отношения только теми записями, которые соответствуют текущей записи в главной таблице. Опция INTEGRITY CASCADE позволяет автоматически удалять соответствующие записи подчиненной таблицы при удалении записи в главной, а INTEGRITY RESTRICTED запрещает удаление подчиненных записей, если существует связанная запись в главной таблице отношения. Реализация механизма ссылочной целостности данных на уровне ядра процессора баз данных Database Engine предоставляет разработчикам все необходимые средства для обеспечения целостности данных в комплексных приложениях, оперирующих с критически важными для заказчиков данными.

Управляемая событиями обработка транзакций

Традиционно в предыдущих DOS-версиях dBASE использовался механизм обработки транзакций, рассчитанный на процедурный язык программирования. Однако для используемой в dBASE для Windows управляемой событиями модели программирования потребовалось разработать новую технологию управления транзакциями. При этом исполняемые в dBASE для Windows транзакции подтверждаются (COMMIT) либо отменяются (ROLLBACK) вне зависимости от места в программе, где они были начаты командой BEGINTRANS.

По мнению автора, выход Windows-версии dBASE означает появление на рынке средств разработки в области баз данных качественно нового продукта, с одной стороны, в полной мере реализующего объектно-ориентированный подход, а с другой — предоставляющего унифицированный доступ к данным и возможность построения приложений по технологии "клиент/сервер". Кроме того, наличие на рынке полностью локализованной версии продукта позволяет говорить о dBASE, как о полном решении для всех категорий пользователей — от офисных систем до разработчиков комплексных приложений.

Borland InterBase 4.0

Вадим Индриков

Если вы пишете свою систему управления базами данных (СУБД), то получиться у вас может все что угодно. Если вы работаете при этом на восьмибитном процессоре и адресуете свое изделие абсолютно неподготовленному пользователю, то получается dBASE. Если все то же самое создается на Apollo Domain или VAX и предназначается для профессионала, то получается персональная СУБД InterBase. Вот так, пожалуй, можно охарактеризовать состояние InterBase, в котором она пребывала в течение долгих лет.

На момент основания InterBase Software (1984 год) Oracle уже пять лет как продавала свои продукты. Но если ровесник InterBase, Sybase, достаточно быстро занял прочные позиции среди мэтров, InterBase долгое время оставался "широко известным в узком кругу". Хотя... Говорить о "неуспехе" СУБД InterBase можно, только если чересчур буквально сравнивать его с остальными серверами данных. Как ни будут, наверное, возмущаться многие из тех, кто давно знает InterBase, я утверждаю, что на протяжении более чем пяти лет InterBase оставался инструментальным пакетом для создания персональных программ. Это на самом деле так, и вся "многопользовательскость" и распределенность его — просто наследование свойств операционных систем, на которых он существовал (Apollo Domain OS, VMS и UNIX). Процессор данных InterBase обладал уникальными свойствами, некоторые из которых таковыми остаются и до сих пор, но основной средой разработки являлись языки C, FORTRAN, Ada и тому подобные. Это, впрочем, не особо смущало создателей САПР и управляющих систем. Оглушительным успехом пользуется он сейчас на платформе NextSTEP.

▲ Новаторские технологии

Грустно рассказывать об этом, поскольку, обогнав многих, InterBase пересидел на старте, и многие вещи перестали быть know-how InterBase.

▲ Одноранговость

InterBase может являться одновременно и сервером, и клиентом по отношению к другим серверам в сети. Анализируя запрос, маршрутизатор InterBase, "Y-valve", определяет: выполнять запрос локально или передать его другому серверу. Это позволяет, не прибегая к разного рода ухищрениям на стороне клиента, обрабатывать в одной транзакции распределенные данные. Двухфазная фиксация транзакций поддерживалась изначально.

▲ Мультипоточность

На мультипоточной Apollo Domain OS и VMS была реализована архитектура "один мультипотоковый сервер — множество клиентов", в то время как стандартный вариант поддерживал доминировавшую на тот момент реализацию "один клиент — один процесс-сервер". Мультипоточность (multithreading) — это возможность выделения внутри одного процесса независимых "подпроцессов", потоков. Последовательности команд, составляющих потоки, могут выполняться независимо друг от друга. Чем это лучше? Ведь и процесс "нормальной" операционной системы может порождать другие процессы. За счет чего может быть достигнута та же степень параллелизма? Дело в том, что потоки работают в одном адресном пространстве, поэтому не требуют тех накладных расходов на организацию обмена данными, каких требуют процессы (каналы, сокеты, разделяемые блоки памяти). Порождение нового потока также проходит живее. Посему потоки часто называют процессами в легком весе. Многопоточные программы требуют меньше ресурсов операционной системы и, в случае серверов данных, оперативнее переключаются с одного клиента на другого. Мультипотоковая архитектура признана сейчас более прогрессивной, чем "классическая". Informix, Oracle и Ingres с большим опозданием воплотили ее в своих последних продуктах.

▲ Экзотические типы данных

Не Informix, как часто утверждают, а InterBase первым реализовал тип данных BLOB. Присвоение полю этого типа позволяет хранить в нем двоичные данные неограниченной длины, будь то графика, звук, видеоизображение и т.д. На базе типа BLOB в InterBase реализован тип "массив". Средства языка InterBase позволяют работать и с тем, и с другим.

▲ UDF

Функции, создаваемые пользователем (UDF), могут вызываться как подпрограммы или использоваться в качестве полноправного участника ограничивающего предложения "WHERE ..." в языке SQL. UDF могут принимать и возвращать великое множество параметров. Учитывая, что функции можно создавать на языке C, Ada и т.д., они могут проводить внутренний анализ содержимого BLOB и выдавать результаты. Хотя передача BLOB в качестве аргумента функции не допускается, возможна передача id этого BLOB, что дает, по сути, тот же эффект. В силу этого возникает принципиальная возможность создавать запросы типа "выбрать фотографии всех граждан с конопатым носом" или "выбрать все песенки, в которых есть упоминания об отдыхе на море". Насколько это реализуемо — уже вопрос соответствующего аналитического аппарата, но не сервера данных.

▲ Транзакции и триггеры

Этим сейчас никого не удивишь, однако InterBase не просто их поддерживает, но делает это весьма привлекательным способом. Транзакции — механизм, позволяющий обеспечивать выполнение группы операций как единого целого и не допустить выполнения ее наполовину. Транзакции в InterBase совершенно независимы — они допускают и вложение, и перекрытие. Один клиент InterBase может работать с множеством транзакций одновременно. Триггер — особый вид UDF, который придается таблице и будет вызван автоматически при модификации ее содержимого. Триггер нельзя обойти. В теле триггера можно проводить разнообразные проверки и в случае необходимости запретить выполнение тех или иных действий. Единственной таблице в InterBase может быть придано более одного триггера, их можно динамически переупорядочивать. Триггеры могут вызываться как перед попыткой модификации записи, так и после того.

▲ Уведомление

Механизм уведомления о событиях позволяет говорить об InterBase как об активном сервере данных. Действительно, клиент может уведомить сервер о своем интересе и заниматься своими делами. По происшествии конкретного события — изменения значения поля, занесения или удаления записи, сервер оповестит об этом.

▲ Создание поколений записей

Любая транзакция, вносящая изменения в базу данных, может привести базу в состояние, при котором в ней содержится противоречивая информация — один счет дебетован, но другой еще не кредитован и т.д. Использование рассогласованных данных, когда один пользователь не закончил транзакцию, а другой уже использует модифицированные ею данные, может привести к печальным результатам. В случае отката транзакции все изменения, ею произведенные, окажутся "писаны вилами по воде". Механизм генерации поколений записей заключается в отслеживании версий записей. Появляется возможность "подсовывать" клиенту, начавшему свою транзакцию в полдень, ту версию базы данных (в согласованном состоянии!), которая существовала в полдень. Зачастую считают, что если сделал один и не сделали другие, то этого и не надо. Ознакомьтесь со стандартом ANSI SQL92 — тогда будет понятно, что уже в середине 80-х InterBase реализовал изоляцию транзакций на уровне 1, 2 и 3 стандарта ANSI 1992 года. Типичные реализации изоляции транзакций основываются на удержании блокировок вплоть до коммита, что не может не сказаться на производительности. Технология, предлагаемая InterBase, полностью исключает возможность возникновения столкновений между "писателями" и "читателями".

▲ Версия 4

В начале 90-х InterBase попал в руки Borland, что и переломило дальнейшую его судьбу. Все больше людей узнают о самом существовании InterBase. Продажи пошли в гору. Совсем недавно вышла новая версия, 4.0.

▲ Новый SQL

Дуализм InterBase заключался в поддержке двух языков программирования — простенького SQL (ANSI 89) и дающего больший простор при последовательной обработке записей GDML. "Четверка" реализовала SQL, совместимый, где это можно, с последними стандартами. Прокручиваемые курсоры позволили отказаться от GDML и унифицировать технику программирования.

▲ Новые платформы

Если до сих пор единственными не-UNIX-платформами являлись VMS и Apollo Domain OS (NextSTEP, несмотря на всю его графику, базируется на ядре Mach 3.0 и отчасти на интерфейсах UNIX BSD), то теперь происходит переориентация на операционные системы с человеческим лицом. Новая версия выпущена пока только для Windows NT и NetWare. Тем, кто

упорно считает, что все, что не UNIX, не дает перспектив в росте, хочу сообщить, что многопроцессорный Tricord опережает RISC UNIX-серверы по абсолютным результатам, а по соотношению "цена/производительность" оставляет их далеко позади. Между тем на нем работает и Windows NT, и NetWare 4.1. А Sequent WinServer 5000 с его 30 процессорами оказывается в далеком отрыве по абсолютным результатам. До сих пор не могу получить внятного ответа ни от одного из фанатиков UNIX — почему, если на машине стоит UNIX (DYNIX/ptx) и она называется Symmetry, то она лучше самой себя (WinServer 5000 — это NT'шный вариант той же железки). Так что если говорить о масштабируемости PC-версий, то ограничения сверху у NT-версии гораздо слабее, чем, например, у HP.

Windows NT поддерживает многопоточность, InterBase это использует. Четверка учитывает сетевые возможности операционной системы и на NetWare позволяет использовать IPX/SPX, а на NT — NetBEUI-транспорт. Единственным протоколом, который он признавал до сих пор, был TCP/IP.

▲ Клиент/сервер

В самом начале я характеризовал InterBase как персональную СУБД. Клиент/сервер все же подразумевает четкое определение спецификаций, согласно которым будут работать в сети серверы и клиенты. Разработанная и воплощенная Borland технология IDAPI внесла ясность в этот темный момент. Выход Borland Database Engine расширил круг средств, при помощи которых можно создавать приложения для InterBase (хотя и не только для него). Дополняющие друг друга IDAPI-драйверы позволяют работать с разнообразными источниками данных, включая серверы данных.

▲ Upsizing!

В новом лозунге Borland гораздо больше истины, чем обычно бывает в рекламных девизах. Дело в том, что для переноса написанных на Paradox или dBASE IV приложений в архитектуру "клиент/сервер" требуется пересмотр технологии программирования. Реляционный сервер принимает запрос на SQL, после чего сам решает, как с ним разбираться. Созданные же при помощи настольных СУБД приложения ориентированы на работу с записями и включают в себя команды навигации по таблицам, выбора того или иного индекса и т.д. При работе по ODBC все эти операции приходится эмулировать. IDAPI изначально был рассчитан не только на связь с серверами данных, но и на поддержку форматов файлов настольных СУБД, что позволяет IDAPI-драйверам более эффективно работать во втором случае. Однако ни один реляционный сервер данных не пустит никого на уровень записи. Поэтому и IDAPI-драйверы эмулируют операции навигации по записям.

Законы жанра заставляют меня на всей этой мрачной картине нарисовать луч света. ExpressLink, IDAPI-драйвер для InterBase, стоит обособлено. Дело в том, что он использует внутренний язык BLR и не является SQL-ориентированным. ExpressLink работает на уровне таблиц, индексов и записей и потому более приближен к технологии IDAPI, ориентированной именно на такой стиль. Ориентация на SQL и ориентация на записи не являются взаимоисключающими: каждая хороша в своем случае. Речь здесь идет о том, что для переноса программы, написанной на Paradox или dBASE, в архитектуру "клиент/сервер" потребуется переделка приложения, иначе на "стыках" будет теряться производительность. Если речь не идет о сервере InterBase. Потому-то я и говорю, что здесь Upsizing означает то, что он означает — увеличение, а отнюдь не перенос. Конечно, и в этом случае переделка потребуется, но не столь значительная.

InterBase может быть использован в аналитических задачах — когда требуется моментально делать срезы по большим объемам данных. Активное ядро InterBase позволяет использовать его в задачах оперативного управления, проведения торгов и т.д. Поддержка массивов и BLOB делают InterBase привлекательным в задачах САПР, картографии, multimedia. Все эти задачи Borland называет OLCP — онлайновая обработка комплексных процессов. Несмотря на то что InterBase не оптимизирован на OLTP (онлайновая обработка транзакций, когда производится множество коротких операций обновления данных), практика показывает, что его с успехом можно использовать и в российских банках. Слабым местом созданных на банковских BTRIEVE систем оказывается не неспособность обрабатывать большое число транзакций в секунду, а слабость в обработке больших запросов. Известны случаи, когда система, обслуживающая порядка 100 клиентов, начинала рушиться (отсоединяя клиентов!) при запуске комплексного фонового запроса.

За рамками данного опуса остался и пакет Delphi — подарок Borland тем, кто все эти годы писал системы обработки данных на Pascal и упорно не уходил с него. Помимо всех прочих прелестей Delphi будет включать в себя локальное, 16-битное ядро InterBase, что позволит разрабатывать приложения на одиноко стоящей машине.

Компания "Весть"
Телефон: (05) 115-97-83
e-mail: bartman@vestjc.msk.su

Компьютерные комплексы «под ключ». Целевые решения по установке, наладке и комплектации компьютерных сетей и офисного оборудования.

Комплексные поставки компьютерных станций. Широкий выбор компьютеров – от серверов до рабочих станций, цветных и черно-белых ноутбуков, модемов и факс-модемов, принтеров, сетевого оборудования, источников бесперебойного питания.

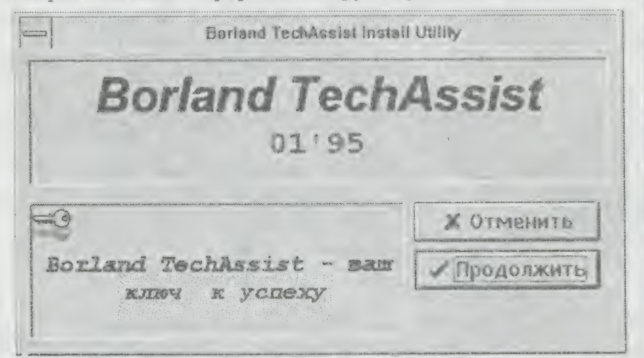


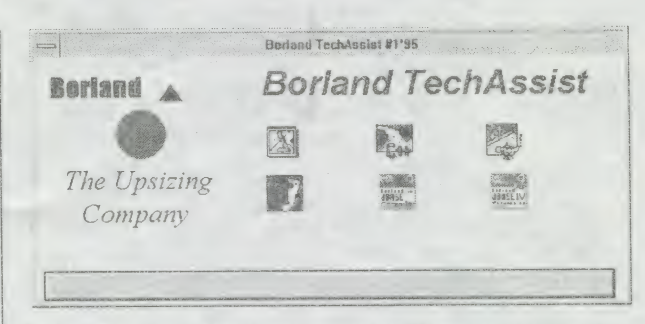
Borland TechAssist

Алексей Федоров

Московское представительство фирмы Borland "Borland AO" выпускает компакт-диск, названный "Borland TechAssist". Диск содержит маркетинговую и техническую информацию, предназначенную для дистрибьюторов и служб технического сопровождения программных продуктов фирмы Borland. Немало интересного найдут на этом диске и пользователи продуктов фирмы Borland: различные программы и исходные тексты с Borland Tech Support BBS, статьи, посвященные продуктам фирмы Borland, собранные из различных периодических изданий, разнообразные информационные материалы и другие полезные сведения, которые несомненно помогут в более успешном освоении программных средств.

Информация, представленная на данном диске, разделена на две категории: маркетинговую и техническую. Маркетинговая информация сгруппирована в нескольких





разделах, среди которых несомненный интерес представляют демонстрационные версии программ, слайд-шоу, иллюстративные материалы, а также сравнительная информация по продуктам других фирм. В разделе технической информации — материалы посвящены различным техническим вопросам применения продуктов фирмы Borland, замечаниям по использованию тех или иных средств, материалам для служб технического сопровождения, а также для подготовки персонала. Последние могут быть использованы в качестве пособий для самостоятельной подготовки или для проведения учебных курсов по ряду продуктов фирмы Borland.

Отличительная особенность данного диска в том, что он создан специально для отечественных пользователей и специалистов и большинство материалов, представленных на нем, приводятся на русском языке (в формате Word for Windows 6.0).

Диск Borland TechAssist планируется выпускать ежеквартально. Распространяться он будет через авторизованных дилеров. Предполагаемая цена одного выпуска — 59 долларов.



Dashboard — приборная панель для Windows

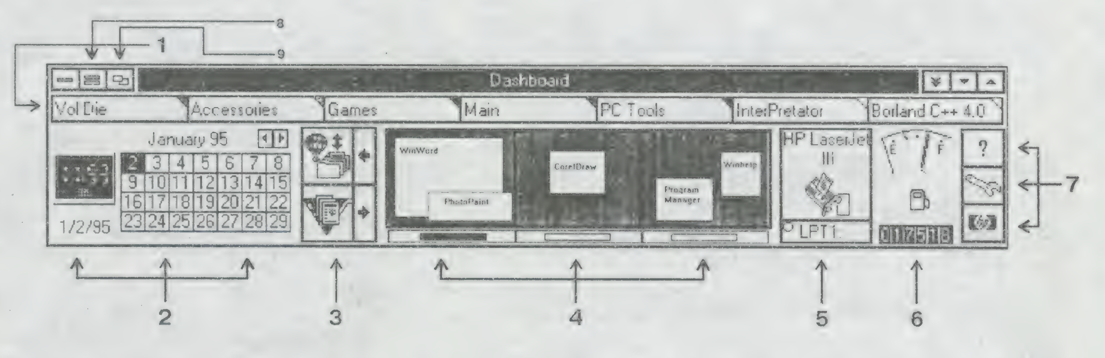
Владислав Головач

▲Введение

Оболочка (программа, загружаемая последней, служащая в основном для запуска других программ) — лицо операционной системы. Стандартная оболочка для Windows — Диспетчер Программ — умеет делать только две вещи: показывать группы со значками и загружать соответствующую программу при щелчке мышью на одном из значков.

Основные недостатки Диспетчера Программ:

- в группу значков нельзя вложить другую группу;
- группе нельзя присвоить собственный значок;
- его трудно расположить на экране так, чтобы он занимал минимум места, но показывал максимум значков;
- он имеет неприятное обыкновение высовываться за край экрана, после чего требуется выравнивание вручную;
- удобный метод "перенеси и оставь" поддерживается в недостаточной степени.



Конечно, если пользоваться Windows эпизодически, необходимость в более удобной оболочке несущественна. Однако если использовать ее постоянно, использование такой оболочки быстро окупается с точки зрения удобства работы.

Многие пользователи даже не подозревают о существовании альтернатив Диспетчеру Программ. Одна из таких альтернатив — пакет Dashboard ("приборная панель"). Его разработала фирма Hewlett-Packard. Во второй половине 1994 года этот пакет был куплен фирмой Borland International.

▲Как выглядит

Окно Dashboard состоит из следующих элементов:

1. Меню Программ (Program Menu). Выполняет ту же функцию, что и Диспетчер Программ в оболочке Windows, а именно: позволяет иметь неограниченное количество групп со значками. В отличие от Диспетчера Программ, позволяет вкладывать в них неограниченное число других групп. Группы здесь реализуют метафору картотеки — набор групп верхнего уровня показывается в виде картотечного ящика; каждая карточка снабжена цветным отогнутым уголком (если у вас цветной дисплей, то это облегчает идентификацию). Если щелкнуть мышью по какой-либо карточке, то она разворачивается и показывает свое содержимое, а при двойном щелчке мыши — раскрывается в окно, похожее на группу Диспетчера Программ (в нем, однако, имеется меню).

2. Часы/Календарь (Clock/Calendar). Встроенный планировщик расписания — одно из главных преимуществ. Чтобы вставить какую-либо запись, достаточно щелкнуть мышью по нужному числу календаря, затем указать время суток.

3. Панель Быстрого Запуска (Quick Launch) — строка инструментов, заполненная наиболее часто используемыми приложениями. Строка инструментов — то же меню, но альтернативы его — не надписи, а значки (возможны и текстовые пояснения); ее можно располагать не только по горизонтали, но и по вертикали.

4. Расширенные экраны (Extended Screens) — позволяют увеличить поверхность вашего рабочего стола без изменения площади монитора или свертывания работающих приложений. Поверхность рабочего стола делится на части, размер которых равен размеру экрана. Таким образом вы видите активную часть, занимающую весь экран, плюс карту всего рабочего стола. Для переключения между областями достаточно дважды щелкнуть мышью по нужной части карты.

5. Диспетчер Печати (Printer Manager) особенно хорош, если у вас несколько принтеров, поскольку позволяет просто перенести файл на нужный принтер.

6. Индикатор Ресурсов (Resources) показывает количество свободной оперативной, дисковой памяти, а также системных ресурсов.

7. Управляющие кнопки:

- вопросительный знак — контекстная справка (при щелчке на нем мышью курсор приобретает вид вопросительного знака. После этого им можно указать на любой элемент панели для получения справки);
- гаечный ключ — кнопка общей настройки (см. ниже);
- третья кнопка — визитка Dashboard'а.

8. Меню задач (Task Menu) — при нажатии этой кнопки выдается список работающих программ. Отсюда вы можете закрыть все программы, а также вызвать окно (Dashboard Run), имитирующее экран DOS, но более удобное. Так, набрав команду DIR, вы получаете список файлов, точно так же, как и в DOS. Но для запуска какой-либо программы не надо набирать ее имя на клавиатуре — достаточно щелкнуть мышью по нужному названию в этом списке.

9. Меню раскладки (Layout Menu) — представьте себе, что вам часто надо запускать несколько программ одновременно, например текстовый процессор, генератор шрифтов и редактор изображений. Для убыстрения такого действия надо лишь открыть эти программы и в этом меню выбрать Save Layout, затем ввести имя созданной раскладки. Имя добавится в список, находящийся здесь же, после чего для одновременного запуска этих программ достаточно будет указать мышью в нужное имя.

Любой элемент панели можно отключить, при этом Меню Программ, Панель Быстрого Запуска и Диспетчер Печати можно отделить от самой доски и расположить в любом месте экрана.

▲Настройка

Процедура настройки довольно проста:

- вы должны щелкнуть мышью по кнопке настройки (средняя управляющая кнопка), после чего появится следующее окно;
- в левой части окна приводится список элементов Dashboard, а справа — варианты настройки. Каждый элемент можно настроить индивидуально.

Например, в окне настройки всей панели вы можете настроить: в Main (Главный) — показывать ли Dashboard постоянно поверх всех окон (Dashboard always in front); использовать ли его как оболочку (Use Dashboard as Windows Shell); заменять ли стандартный Диспетчер Задач (Replace Windows Task Manager); показывать ли значки в полуоткрытой карточке группы (Show icons in menus and screens). Также вы можете указать, что показывать в заголовке окна Dashboard'а — время/дату, количество свободных ресурсов или ничего; это относится и к значку свернутого окна Dashboard.

В самом низу окна находится строчка с выбором, как расположить панель на экране — вертикально или горизонтально.

В Toolbars (Панели инструментов) можно установить, какие элементы панели надо отделить от главной панели и нужно ли их сворачивать вместе с Dashboard.

▲Заключение

Важной особенностью Dashboard является то, что он полностью совместим с Диспетчером Программ. Он, например, использует его групповые файлы (.GRP). Это означает, что любая программа, изменяющая группы Диспетчера Программ, изменяет и структуру меню. В частности, подобное поведение имеет то преимущество, что даже в Диспетчере Программ позволяет использовать подгруппы. Например, связав расширение .GRP с программой DASHGRP.EXE (программа показа групп), вы можете создать в какой-либо группе Диспетчера Программ новый элемент с этим расширением файла.

Интересно, что при заметно более богатых и удобных функциях (по сравнению с Диспетчером Программ) Dashboard очень эффективно использует оперативную память (путем соответствующей настройки можно добиться, чтобы он занимал меньше памяти, чем Диспетчер Программ). Поэтому его разумно использовать даже на маломощных компьютерах, не опасаясь потерь производительности.

Если же у вас избыток памяти, возможно, стоит обратить внимание на другие, более мощные оболочки.

Операционная система OS/2 фирмы IBM до последнего времени оставалась для нас достаточно экзотическим предметом. В России с ней работал лишь ограниченный круг профессионалов, все остальные довольствовались слухами...

История OS/2

Камилл Ахметов

Вплоть до 1990 года мы твердо знали, что наследницей системы DOS станет OS/2. С выходом Microsoft Windows 3.0 оказалось, что это не очевидно. Сейчас, когда Microsoft делает все, чтобы осенью 1995 года состоялся переход возможно большего числа пользователей к системе Windows 95, на рынке реально существует мощная 32-разрядная система нового поколения IBM OS/2 Warp 3, которая уже приступила к сбору урожая на угодьях Microsoft.

А дело было так...

По дороге к операционной системе 90-х

Преобладающим типом компьютеров на массовом рынке IBM PC-совместимых ПК к 1987 году были компьютеры на базе 16-разрядного микропроцессора 80286, 32-разрядный 80386 только входил в моду. 80286 мог работать в защищенном режиме, 24-разрядная адресная шина позволяла ему использовать до 16 Мбайт оперативной памяти. Появившиеся в 1984 году компьютеры на базе 286-го микропроцессора были переходным явлением, когда продолжать развивать рынок компьютеров на базе морально устаревшего 8086 было уже нельзя, а по-настоящему передовое решение еще не было отработано. Наверное, поэтому 286-й так долго был популярен в стране переходных явлений, которую мы с вами населяем...

Операционная система DOS версии 3.3 (тоже, кстати, очень долго бывшая популярной у нас) была переходным явлением другого рода. Весной 1987 года DOS 3.3 была нужна для того, чтобы с новыми компьютерами семейства IBM PS/2 поставлялась новая операционная система. В действительности DOS 3.3 можно было лишь формально назвать «новой» — это было очередное усовершенствование все той же DOS, она отличалась поддержкой 3-дюймовых дискет на 1,44 Мбайт и разделов жесткого диска объемом 32 Мбайт.

Целью системы OS/2 было использование возможностей передовых микропроцессоров, она не должна была работать на машинах с микропроцессором 8086. Первая планка, которую должны были взять изготовители OS/2 — IBM и Microsoft, — использование всей адресуемой памяти, защищенного режима и многозадачных возможностей микропроцессора 80286. Для внедрения на существующий рынок программного обеспечения, разумеется, необходима была совместимость с DOS 3.3.

Продажи новой операционной системы начались в ноябре 1987 года. Microsoft поставляла MS-OS/2, работавшую на любых IBM PC-совместимых персональных компьютерах; IBM OS/2 работала только на компьютерах IBM PS/2 с новой шиной MCA. На самом деле существовало даже три варианта OS/2 1.0 — MS-OS/2 1.0, IBM OS/2 1.0 SE (Standard Edition, стандартная редакция), IBM OS/2 1.0 EE (Extended Edition, расширенная редакция). Расширенная редакция, адресованная профессиональным пользователям, содержала телекоммуникационную программу Communication Manager и систему управления базами данных Database Manager.

Реально OS/2 1.0 предлагала конечным пользователям следующее:

- использование защищенного режима микропроцессора 80286 и до 16 Мбайт адресуемой памяти;
- вытесняющая (preemptive) многозадачность с одновременным выполнением до 12 программ;
- новый командный процессор;
- различные системные средства.

Требование о совместимости с приложениями для DOS было выполнено не до конца, некоторые программы для DOS не выполнялись. Одновременно могла работать только одна DOS-задача в специальной «совместимой» сессии OS/2 (compatibility box), эмулирующей реальный режим микропроцессора 8086,— ей было доступно 640 Кбайт памяти. OS/2 не баловала разнообразием драйверов устройств ввода-вывода, а системные требования (до 3 Мбайт оперативной памяти) для того времени были слишком высоки.

Странная история системы OS/2 и оболочки Windows

Важной вехой в истории IBM OS/2 и MS-OS/2 была версия 1.1 1988 года, также существовавшая в трех редакциях. Ее основное отличие — графическая многооконная среда Presentation Manager. Графический интерфейс PM был не просто оболочкой операционной системы —

Многозадачность и... многозадачность

Чем же отличается многозадачность OS/2 от многозадачности Windows 3.x? OS/2 реализует режим preemptive multitasking, также называемый time-sliced multitasking. Это значит — *вытесняющая многозадачность*, или многозадачность с разделением времени. При работе в режиме вытесняющей многозадачности операционная система отводит каждому параллельно выполняемому приложению строго определенный промежуток времени. В этом случае такая система, как OS/2 (или, например, Microsoft Windows NT), сама «замораживает» одну задачу и активизирует следующую, при этом создается полное впечатление одновременной работы программ.

В системе Microsoft Windows 3.x основным многозадачным режимом является режим *кооперативной многозадачности*, в котором функционируют Windows-приложения. При кооперативной многозадачности текущая программа захватывает столько процессорного времени, сколько ей необходимо, а программы, выполняемые в фоновом режиме, таким же «кооперативным» путем делят между собой остаток времени. Именно этим объясняются ощутимые перерывы в выполнении фоновых задач Windows. Кроме того, в 386-м Расширенном режиме Windows 3.x реализована работа с разделением времени для DOS-приложений.

он имел собственный API (интерфейс прикладного программирования), причем значительно более богатый, чем API программ символьного режима. Программы, пользующиеся для взаимодействия с пользователем вызовами API OS/2 Presentation Manager, работали в графическом интерфейсе PM.

Напомним, что фирма Microsoft начала самостоятельные разработки в области графического интерфейса пользователя еще в 1983 году, и первая версия графической среды Microsoft Windows (для DOS) появилась в 1985 году. А в 1988 году была выпущена Windows/286 (версия 2.11), с надписью на коробке «Microsoft Windows/286 *Presentation Manager*». Эта оболочка также имела собственный API для работы с прикладными программами, но могла использовать для работы прикладных программ только нижние 640 Кбайт памяти DOS и Expanded-память.

В 1989 году появились IBM OS/2 1.2 и MS-OS/2 1.21. Графический интерфейс Presentation Manager этих версий был переработан и стал «трехмерным». В OS/2 появилось новое приложение для работы с файлами — File Manager. Главное достижение OS/2 1.2 — опциональная поддержка новой файловой системы HPFS — High Performance File System. Преимущества высокопроизводительной файловой системы особенно важны для серверов и работы с серьезными базами данных. Важным свойством HPFS была поддержка длинных имен файлов. Впрочем, конечным пользователям до сих пор мало что известно об HPFS, поскольку на диск, отформатированный под HPFS, нельзя установить DOS и Windows.

На 1990 год разработчиками был запланирован выпуск 32-разрядной версии OS/2 2.0 для компьютеров с микропроцессорами не ниже 80386. Фирма Microsoft параллельно вела второй проект — разработку новой версии Windows, графический интерфейс пользователя этой среды в основном соответствовал виду OS/2 Presentation Manager. В конце 1989 года фирма Microsoft еще заявляла, что будущая OS/2 станет лучшей операционной системой, чем DOS, и лучшей оконной средой, чем Windows. Но тогда же в печати появилось мнение, что Microsoft делает далеко не все для развития OS/2 и сосредоточена именно на Windows.

Что ж, рыночное чутье фирмы Microsoft всегда было на высочайшем уровне, а рыночная ситуация для OS/2 не стала намного благоприятней. Фактическим стандартом оставались компьютеры для DOS, программы для DOS и (даже после выпуска DOS версии 4) DOS 3.3. Для работы подавляющего большинства массовых прикладных программ хватало 640 Кбайт оперативной памяти. Вспомните наиболее впечатляющие программные достижения того времени — текстовый процессор Microsoft Word 5, базы данных dBase III Plus фирмы Ashton-Tate и Paradox 3 фирмы Borland, классическую электронную таблицу Lotus 1-2-3, и, конечно, компиляторы фирмы Borland — Turbo Pascal и Turbo C... Все эти программы работали под DOS, прекрасно функционировали на 286-х машинах и обеспечивали вас всем, что вы могли пожелать. Такие новации, как Word 5.5, dBase IV и Turbo C++ 1.0, воспринимались сдержанно и осторожно. Ну, а переход на новую операционную систему, для которого почти неминуемо требовались дополнительные вложения в наращивание аппаратной мощности, и вовсе казался авантюрой.

С другой стороны, три года непрерывной рекламы OS/2 как преемницы DOS привели к некоторому результату — общественное мнение было готово к появлению чего-то, что придет на смену DOS. Новая операционная система должна была иметь красивый и удобный «трехмерный» интерфейс, передовые методы управления памятью, многозадачные возможности... и не быть OS/2.

В 1990 году была выпущена OS/2 1.3, которая имела повышенное быстродействие, улучшенную, хотя все же несовершенную под-

держку DOS-приложений и требовала для работы всего 2 Мбайт памяти. Был расширен набор драйверов ввода-вывода. Важным шагом было включение в OS/2 системы масштабирования PostScript-шрифтов Adobe Type Manager, традиционного средства для профессиональной издательской деятельности.

Но основным событием 1990 года стал выпуск Microsoft Windows 3.0. На 286-х машинах Windows могла использовать всю доступную оперативную память, каковой требовался всего 1 Мбайт (Стандартный режим). Была в Windows и многозадачность — правда, не вытесняющая, как в OS/2, а «кооперативная». Использование виртуальной памяти было возможно только при работе Windows 3.0 с 32-разрядными микропроцессорами, то есть 80386 и выше, при наличии не менее 2 Мбайт памяти (386-й Расширенный режим). А еще Windows по-прежнему могла работать на машинах класса XT с использованием только нижней памяти (Реальный режим).

Таким образом, Windows обладала всеми желаемыми признаками, кроме одного — она не была операционной системой. Windows достраивала ядро и командный процессор системы DOS собственными ядром и графическим интерфейсом. С DOS-приложениями под Windows не было существенных проблем. Приложения, написанные для Windows, взаимодействовали с ее графическим интерфейсом, пользовались сервисом Windows и всей доступной для Windows-приложений памятью.

Ну кто тут будет разбираться, «настоящая» Windows операционная система или нет? Она может все, что требуется, и дело с концом! В считанные месяцы программистские фирмы перешли на производство программ для графического интерфейса Windows. С этого момента фирма Microsoft окончательно сконцентрировалась на собственной концепции «операционной системы 90-х» и перестала заниматься совместными разработками с IBM.

За один год поставок Windows 3.0 Microsoft продала 5 миллионов экземпляров продукта, и это было лишь началом грандиозного успеха. В планах фирмы стояли поддержка и развитие OS/2 (без IBM) и создание новой операционной системы NT (New Technology). Ближе к делу оказалось, что запланированная на выпуск в 1992 году система OS/2 3.0 — это и есть NT. Забегая вперед, скажем, что выпущенная в конце концов осенью 1993-го года новая операционная система получила наименование... *Windows NT.*

Вряд ли имеет смысл сейчас спорить о том, последовательна или непоследовательна в своих решениях была фирма Microsoft. Стандарт на графический интерфейс для IBM PC-совместимых компьютеров в конечном счете ввела именно Microsoft, это фактический результат ее деятельности. Важно и то, что требования к аппаратной части повышались Microsoft не скачкообразно, а шаг за шагом.

Итак, вместо проверенного союзника на рынке операционных систем IBM вдруг обзавелась сильнейшим соперником, моментально ушедшим далеко вперед. Общий объем продаж первого поколения OS/2 не достигал и полумиллиона, а тут еще надо поддерживать новый стандарт — Windows, и при этом предложить клиентам больше и лучше, чем они получали от Microsoft.

В этих достаточно трудных условиях и состоялось второе рождение OS/2 — операционной системы от IBM.

OS/2 без Microsoft — минус на минус дает плюс

Они появились практически одновременно, в апреле 1992 года —

IBM OS/2 2.0 и Microsoft Windows 3.1. Новая операционная система от Microsoft была именно тем, чего ждал рынок,— улучшенный интерфейс, drag-and-drop, поддержка мультимедиа, новые издательские возможности, полная совместимость с существующими приложениями для Windows 3.0 и... ничего принципиально нового. Windows осталась 16-разрядной системой и предлагала два режима работы — Стандартный и 386-й Расширенный. Во-первых, в мире было еще полно 286-х компьютеров. Во-вторых, 32-разрядная система, когда-то заявленная Microsoft на 1992 год, как бы она ни называлась, еще далеко не была готова.

В момент появления IBM OS/2 2.0 многие из нас не могли и мечтать о том, чтобы использовать ее — для работы требовался микропроцессор не ниже 80386 и не менее 4 Мбайт памяти. За что такая цена? Новая версия OS/2 была целиком 32-разрядной — ядро, драйверы устройств и система управления памятью использовали преимущества 32-разрядной организации микропроцессоров, совместимых с 80386. Это обеспечило высокую производительность работы всей системы. Был сильно усовершенствован механизм поддержки многопоточных (multi-threaded) процессов.

Весьма хорош был объектно-ориентированный графический интерфейс пользователя OS/2 Workplace Shell. Модель SOM (System Object Model), на которой он был основан, являлась объектной моделью, не зависящей от языка программирования. Разумеется, OS/2 2.0 работала на любых IBM PC-совместимых компьютерах, а не только на машинах производства IBM.

Указанные черты новой версии ободрили пользователей OS/2 со стажем и привлекли к ней внимание изрядного количества профессионалов, действительно озабоченных тем, что DOS и Windows используют не все возможности их 386-х и 486-х компьютеров. Однако рынок DOS- и Windows-приложений был на порядок больше. Что же сделала IBM в OS/2 для поддержки программ для DOS и Windows?

Специально для работы с DOS- и Windows-программами в OS/2 2.0 появился режим виртуальных DOS-машин (VDM). В одном сеансе OS/2 теперь могло работать до 240 VDM, с полностью индивидуальной настройкой параметров CONFIG.SYS и AUTOEXEC.BAT и собственными приоритетами. Для работы с Windows-программами в VDM задействовалось расширение WIN-OS2, включающее лицензионный код Windows 3.0. Параллельно выполняемые VDM с сеансами DOS и программами Windows работают, естественно, в режиме «настоящей» многозадачности OS/2, а не кооперативной многозадачности Windows.

Однако проблемы оставались, и довольно многочисленные. Не все DOS- и Windows-приложения стабильно работали под OS/2 2.0. Многие отметили, что встроенная поддержка Windows 3.0 довольно медлительна, отдельной проблемой было отсутствие возможности работы в 386-м Расширенном режиме Windows. Кроме того, IBM явно не ожидала настолько быстрого перехода пользователей к Windows 3.1 и технологии масштабирования TrueType. С 1992 года развитие рынка Windows фактически относилось уже к Windows 3.1, все больше популярных приложений работали под Windows 3.1, но не под 3.0.

Были и другие недочеты. Скажем, не так быстро, как хотелось бы, шло обогащение системы 32-разрядными драйверами устройств ввода-вывода. По этой причине, например, на большинстве видеоадаптеров приходилось работать с Windows-программами на полном экране Windows, а не в окне Workplace Shell.

1993 год был отмечен появлением новых версий Windows и OS/2. OS/2 2.1 включала лицензионный код Windows 3.1 и поддерживала 386-й Расширенный режим работы. Набор драйверов устройств был увеличен. Были также исправлены ошибки предыдущей версии, работа DOS- и Windows-программ под OS/2 2.1 стала стабильной, и, таким образом, проблемы с работой Windows-приложений под OS/2 были фактически решены. Нельзя не заметить, однако, что OS/2 2.x, в отличие от Windows, требовала для нормальной работы все же не 4 Мбайт памяти, а хотя бы 6 (лучше 8). Принципиальным новшеством OS/2 2.1 было ее мультимедиа-расширение.

Интересным рыночным ходом был выпуск версии OS/2 2.1 «для Windows». В комплект поставки OS/2 для Windows включалась только часть WIN-OS2, ответственная за интерфейс между Windows и OS/2, а собственно код Windows 3.1 — нет. В этом случае для работы в Windows в OS/2 «вживлялся» код системы Windows 3.1, установленный на компьютер. Версии Windows 3.11 и Windows 3.11 для рабочих групп были полностью совместимы с Windows 3.1, и поэтому претензий к актуальности поддерживаемой версии Windows не возникло.

Если OS/2 2.0 явилась аргументом для наиболее яростных скептиков, считавших, что IBM вообще неспособна выпустить приличную операционную систему для собственного персонального компьютера, то OS/2 2.1 переубедила тех, кто считал, что OS/2 никогда не будет хорошо продаваться. Темпы продаж OS/2 2.1 реально выросли, а объем поставок к осени 1994 года достиг 5 миллионов.

Другое дело, что объем продаж Microsoft Windows в то же самое время уже превышал 50 миллионов...

▲Tide is turning

Если бы Microsoft выпустила Windows 4.0 к концу 1994 года, то,

скорее всего, IBM просто не смогла бы в приемлемые сроки противопоставить ей что-либо столь же впечатляющее и притом совместимое. Но Windows 4.0 превратилась в Windows 95. Это чтобы в наше-то время люди смогли прожить целый год совсем без новой операционной системы!

И опять — все ждали появления новой операционной системы, красивой, устойчивой, возможно более «легкой» и конечно же 32-разрядной — но при этом максимально совместимой со всеми существующими приложениями. Правда, такой системы ждали от Microsoft. Но фирма IBM подошла к сложившейся ситуации спокойно и со знанием дела.

К осени 1994 года было завершено бета-тестирование OS/2 2.99 для Windows, в проекте известной также под именем Personal OS/2. Начнем с того, что эта система действительно работает (а не прохлаждается) на компьютерах с 4 Мбайт памяти, разрабочики специально оптимизировали ее код для продуктивной работы с малым объемом физической памяти. Графический интерфейс Workplace Shell был еще более усовершенствован, а конфигурирование системы стало значительно проще и удобнее. Из всех версий OS/2 она самая «совместимая» — WIN-OS2 теперь работает со всеми версиями Windows 3.1x и Windows 3.1x для рабочих групп, а при установленной поддержке Microsoft Win32s — с 32-разрядными приложениями для Win32s, за исключением многопоточных.

11 октября 1994 года новая версия OS/2 была выпущена. IBM сделала это, не дожидаясь выставки COMDEX/Fall'94, за месяц до ее начала. В официальное наименование OS/2 3.0 вошло одно из ее прозвищ — «Warp» — «сверхсветовая скорость» (из фильма «Star Trek»).

В ближайшие месяцы станет окончательно ясно, сколько новых пользователей сможет привлечь OS/2 Warp 3, «третья сверхсветовая», и какую роль она окажет на разработчиков программного обеспечения. Наверняка многим квалифицированным пользователям захочется хотя бы попробовать поработать с широко разрекламированной новинкой. Конкретно в России многое будет зависеть от выхода локализованной версии OS/2 Warp.

Конечно, IBM старается внедрять свою идеологию не только для PC — операционная система для PowerPC от IBM уже готова. Сейчас у фирмы IBM есть еще несколько месяцев, в течение которых Microsoft не сможет ей «мешать» напрямую. Microsoft, разумеется, будет тем временем делать Windows 95 максимально выигрышной, имея пример не призрачного, в лице бета-версии, а вполне реального соперника.

Excel 5.0: заметки на полях

Часть первая, неформальная

Валерий Очков

Решатель электронных таблиц Excel: взгляд шутника, дериватора и эстета

Рассказывают такую историю.

Воздушный шар вырвался из облаков. Летящие увидели на земле человека и закричали: *"Где мы находимся?!"* — *"Вы находитесь в корзине воздушного шара"*, — таков был ответ. Его дал *математик*. Только от математика можно услышать совершенно точный и совершенно бесполезный ответ.

Статьи, просто информирующие читателя о новых программных продуктах, подобны ответу математика если не своей точностью, то своей бесполезностью. Для серьезного разговора места в журнале мало. Часто вместе со статьей выходит новая версия программы с новыми возможностями, требующая нового анализа (феномен света угасшей звезды).

Есть особый род **шутников** (а к ним относит себя и автор), получающих удовольствие от попыток розыгрыша компьютера. "Много ты, машина, о себе воображаешь. Посмотрим-ка, проглотишь ли ты вот это!?" — так или примерно так думают они, вводя в ЭВМ заведомо "неправильную" информацию. Анализ реакции компьютера на подобные шутки пользователя может дать не меньше, чем классический разбор программы.

Автор надеется, что читатель уже заинтригован названием и началом статьи. Смысл термина *дериватор* в ней зашифрован. Теперь несколько слов о ее теме.

В статье будет рассмотрено решение в среде электронных таблиц Microsoft Excel 5.0 (русская версия) семи задач из области вычислительной математики. Примеры взяты не из пакета Excel 5.0 (а там есть схожие задачи), а из других источников [1-3] и подобраны так, чтобы можно было показать не легкость и изящество их решения, а возникающие при этом осложнения и пути их преодоления (да простят автора за это сотрудники московского отделения Microsoft, предоставившие ему для опробования экземпляр Excel 5.0). К примерам, прилагаемым к пакетам фирмами-разработчиками, следует относиться с большой опаской. Тут можно вспомнить эпохальную машину "Наири-2", в матобеспечении которой была программа "Поиск минимума функций". Так вот, после опробования в ее названии слово "функций" приходилось менять на слово "функции", так как минимум искался только у встроенной в программу функции и больше ни у какой другой.

Автор никогда бы не осмелился так "отблагодарить" фирму Microsoft, если б не тот факт, что Решатель (Solver), интегрированный в пакет Excel и реализующий технологию *"Что-если"*, фирма Microsoft приобрела у сторонних организаций — у фирм *Frontline System* и *Optimals Methods*.

Пример 1

Поиск корня уравнения

Решение с помощью команды *Подбор параметра...* в меню *Сервис*		
Начальное значение X и корень	*1.732161*	В ячейке B4 было 3, должно быть 1,732051...
Функция	0.000383	В ячейке B5 формула B4^2-3
Решение с помощью команды *Подбор параметра...* в меню *Сервис*		
Начальное значение X и корень	*3*	В ячейке B8 было 3, должно быть 1,732051...
Функция	0.000600	В ячейке B9 формула (B8^2-3)*0,0001
Решение с помощью команды *Поиск решения...* в меню *Сервис*		
Начальное значение X и корень	*1.730923*	В ячейке B12 было 3, должно быть 1,732051...
Функция	0.000000	В ячейке B13 формула (B12^2-3)*0,0001
Метод секущих		Точность 0,1 - корень 1,8
		Точность 0,01 - корень 1,721529
		Точность 0,001 - корень см. в B12

Рис. 1

Пример 1. Поиск корня алгебраического уравнения

Есть хорошее правило — при отладке и испытании новой программы нужно подсунуть ей задачу, решение которой заранее известно.

Рис. 1 — это таблица с результатами поиска корня простейшего уравнения X2 — 3 = 0 командой *Подбор параметра...* из меню

Сервис. После ее отдачи на дисплей выводится диалоговое окно (рис. 2), в котором пользователь заполняет три поля: *"Установить в ячейке"* (указывается ячейка B5), *"Значение"* (0) и *"Изменяя ячейку"* (B4). Четвертый установочный параметр — исходное значение *X* (там было 3), от которого начинается поиск корня, записывается в ячейке B4 самой электронной таблицы. В таблицах, иллюстрирующих статью, изменяемые пользователем (исходные) величины отмечены курсивом. Полужирный шрифт отмечает комментарии.

Из рис. 1 видно (см. строки 2-5), что команда *Подбор параметра...* сработала правильно.

В записках Сергея Довлатова отмечено, что *семья — это когда по шуму в ванной можно догадаться, кто там моется*. К сожалению, ни в Help'е, ни в руководстве пользователя пакета Excel ничего не сказано о том, какие вычислительные методы заложены в Решатель. Приходится догадываться по *"шуму в ванной"* — по результатам тестирования команды *Подбор параметра...* через изменение тройки входных параметров: вид анализируемого уравнения, значение начального приближения (у нас оно равно трем) и точность расчета. Последний входной параметр пользователь менять не может, что чревато досадными ошибками: если в нашей задаче о корне уравнения X2 — 3 = 0 умножить его левую часть на коэффициент 0,0001 (см. строки 7-9 на рис. 1), то искомый корень "размажется" на недопустимо широком интервале оси X, и никакая сила не заставит машину найти нужное решение — плюс-минус квадратный корень из трех.

В описанной простейшей задаче кроется подвох, связанный с разным подходом к ней прикладных математиков и специалистов, использующих численные методы для решения своих прикладных проблем. Первые связывают точность расчетов с величиной функции. Вторым же на функцию наплевать (они часто даже не представляют, в каком диапазоне она меняется). Главное, чтобы достигнутое расчетом значение аргумента находилось как можно ближе к тому месту, где функция меняет свой знак, то есть к корню уравнения. Из-за этого нередко приходится отказываться от стандартных методов вычислительной математики и "изобретать велосипед" — писать свою собственную программу поиска корня методом секущих, Ньютона, половинного деления и т.д., ориентируясь на свойства конкретной моделируемой системы.

```
Function Y(X)
    Y = X ^ 2 - 3 ' Анализируемое уравнение
End Function

Sub Кнопка1_Щелкнуть() ' Процедура "Метод секущих" (шум в ванной)
    TOL = ,0001 ' Точность расчета
    X = [B4] ' Первое приближение к корню
    If Abs(Y(X)) < TOL Then
        Root = X
       Else
        If X = 0 Then h = TOL Else h = X * TOL
        X0 = X: X1 = X + h
        Do ' Цикл приближения к корню
            Root = X1 - Y(X1) * (X1 - X0) / (Y(X1) - Y(X0))
            X0 = X1: X1 = Root ' Подготовка к следующему приближению
        Loop Until Abs(Y(X1)) <= TOL
    End If
    [B4] = Root ' Корень
End Sub
```

Изменение точности расчета (и многое другое) допустимо при использовании второй команды Решателя — команды *Поиск решения..*, после отдачи которой появляется одноименное диалоговое окно — см. рис. 3. Эта команда намного перекрывает первую команду — команду *Подбор параметра..*, которая лишняя и оставлена в меню *Сервис*, по-видимому, для совместимости со старыми версиями Excel'а. Работая с командой *Поиск решения..*, можно менять не только точность, но и другие параметры поиска — см. рис. 4. Но опять же процесс приближений к корню ориентирован не на значение аргумента, а на значение функции, что чревато новыми ошибками.

С поиском корня команда *Поиск решения...* справилась, посмотрим, "проглотит" ли она оптимизационные задачи?!

Подбор параметра

Установить в ячейке: B5 — OK
Значение: 0 — Отмена
Изменяя ячейку: B4 — Справка

Рис. 2

Примеры 2-4. Задача старушки "Наири": поиск минимума функции

В специальной литературе приводятся три тестовые функции (функция Розенброка, функция Пауэлла

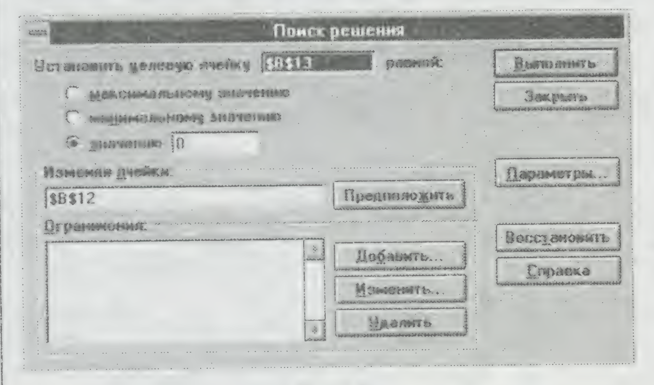


Рис. 3

Параметры поиска решения

Максимальное время: 100 секунд
Итерации: 100
Точность: 0,001
Допустимое отклонение: 5 %
Линейная модель
Показывать результаты итераций
Автоматическое масштабирование
Оценка: линейная / квадратичная
Производные: прямые / центральные
Метод: Ньютона / сопряженных градиентов
OK | Отмена | Загрузить модель... | Сохранить модель... | Справка

Рис. 4

Не то беда, Авдей Флюгарин,
Что родом ты не русский барин,
Что на Парнасе ты цыган...
Беда, что скучен твой роман.

Не то беда, что ты, Решатель, не решил простейшую оптимизационную задачу. Более мощные специализированные пакеты часто тут оказываются бессильными. Нет и не может быть универсальной программы-панацеи, годной на все случаи. Беда, что неверный ответ сопровождается дезинформирующим сообщением.

	Пример 3	
Поиск минимума функции Пауэлла		
		Примечания
Значение X1	*0.0006649*	В ячейке B4 было 0, должно быть 0
Значение X2	*-0.0000673*	В ячейке B5 было 0, должно быть 0
Значение X3	*-0.0000721*	В ячейке B6 было 0, должно быть 0
Значение X4	*0.0022141*	В ячейке B7 было 10, должно быть 0
Значение функции	0.000000000	Должно быть 0
В ячейке B8 формула (B4+10*B5)^2+5*(B6-B7)^4+(B5-2*B6)^4+10*(B4-B7)^4		
Ответ найден за две попытки		

Рис. 6

и двумерная экспоненциальная функция), с которых рекомендуют начинать испытания программ поиска минимума [1-3]. Если компьютер эту тройку "разъяснит", то можно... проводить следующие испытания. Если же проверка завершится неудачей, то программа отбраковывается. С

	Пример 2	
Поиск минимума функции Розенброка		
		Примечания
Значение X	*1.000003909*	В ячейке B4 было 10, должно быть 1
Значение Y	*1*	В ячейке B5 было 10, должно быть 1
Значение функции	0.0000000	Должно быть 0
В ячейке B6 формула 100*(B5-B4^2)^2+(1-B4)^2		
Ответ найден за две попытки		

Рис. 5

функциями Розенброка и Пауэлла (см. рис. 5 и 6) Решатель справится, но за две попытки: сначала выдается неверный результат, после чего пользователь, знающий правильный ответ (функция-то тестовая), должен еще раз отдать команду *Поиск решения..*, но уже от новой начальной точки — от ложного ответа первого приближения. Попытки автора за счет изменения параметров поиска (максимальное время счета, максимальное число итерации, точность и т.д. — см. рис. 4) избежать "привала" на ложном решении не привели к успеху. Ничего не было бы страшного в таком своеобразном двухэтапном (с привалом) процессе, если бы Решатель не выдавал пользователю ложную информацию о том, что после первого этапа *"Решение найдено, все ограничения выполнены"*. Тут вспоминается знаменитая эпиграмма А.С.Пушкина на Фаддея Булгарина:

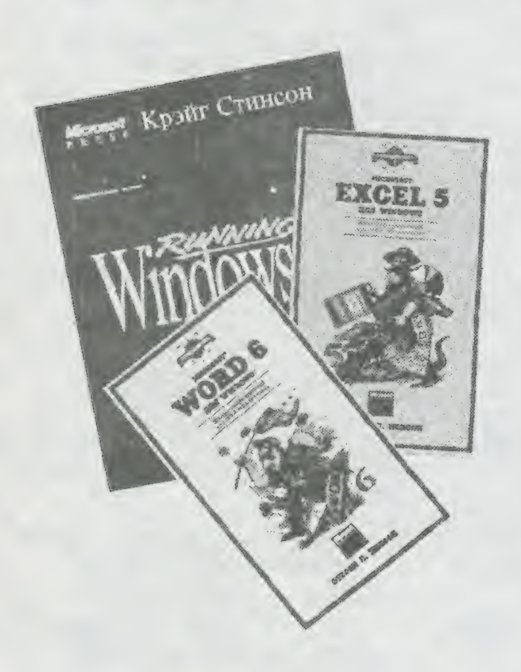
А вот с двухмерной экспоненциальной функцией (рис. 7) вышел полный конфуз. Чья здесь вина — пакета Excel или автора, не сумевшего подобрать соответствующие параметры поиска, трудно сказать. Но факт остается фактом — задача осталась нерешенной. Автор даже не хотел помещать этот пример в статье ("слово — серебро, молчание — золото"), но три фактора заставили поменять решение. Во-первых, Excel Solver — это детище не Microsoft, а другой фирмы, и мы уже это отмечали. Во-вторых, московская группа поддержки Excel, куда обратился автор, не дала ответа, а переправила примеры дальше в Мюнхен — в европейский филиал Microsoft.

	Пример 4	
	Поиск минимума двумерной экспоненциальной функции	
		Примечания
Значение X	*1.092146573*	В ячейке B4 было 100, должно быть 1
Значение Y	*198*	В ячейке B5 было 100, должно быть 10
Значение функции	0.149845	Должно быть 0
В ячейке B6 вызов функци *ДвуЭксп* из модуля		
	Задача не решается	
' Двумерная экспоненциальная функция		
Function ДвуЭксп(X; Y)		Метод "Два шага"
ДвуЭксп = 0		
For A = ,1 To 1 Step ,1		
ДвуЭксп = ДвуЭксп + (Exp(-A * X) - Exp(-A * Y) - Exp(-A) + Exp(-10 * A)) ^ 2		
Next		
End Function		

Рис. 7

ПЕРЕВОДЫ БЕСТСЕЛЛЕРОВ

Microsoft*Press*

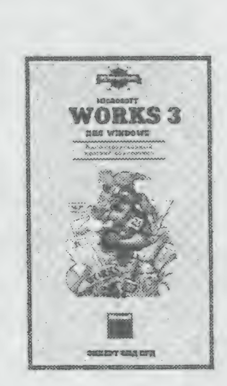


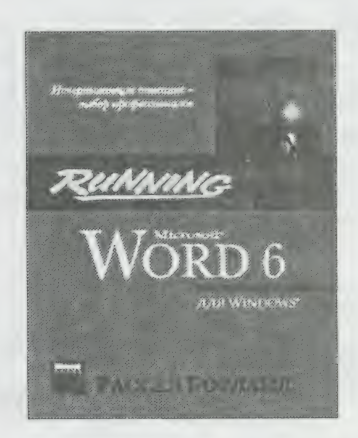
Первые книги издательства:

Крэйг Стинсон ***"Running Microsoft Windows 3.1"***

Стефен Нелсон ***"Путеводитель по Microsoft Word 6 для Windows"***

Стефен Нелсон ***"Путеводитель по Microsoft Excel 5 для Windows"***





Новинки:

Рассел Борланд ***"Running Microsoft Word 6 для Windows";***

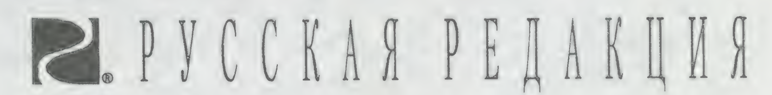
Зихерт и Вуд ***"Путеводитель по Microsoft Works 3 для Windows";***

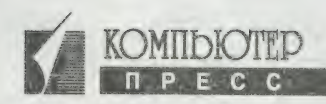
Готовятся к публикации:

Росс Нелсон ***"Running Microsoft Visual Basic 3 для Windows";***
Стефен Нелсон ***"Путеводитель по Microsoft Access 2 для Windows";***
Microsoft Press ***"Толковый словарь по вычислительной технике";***
Cobb Group ***"Running Microsoft Excel 5 для Windows"***

Приглашаем к сотрудничеству региональных дилеров

ИЗДАТЕЛЬСКИЙ ОТДЕЛ
ТОО "CHANNEL TRADING Ltd."
тел. (095) 142-0571





Из Мюнхена пришел ответ, смысл которого заключался в том, что это не вина пакета Excel, а вина... функции — не нужно быть такой сложной и противной. В-третьих, автор надеется, что кто-то из читателей в этой проблеме разберется и даст правильный ответ. А пока помещаем ниже простейшую Basic-процедуру, взятую из [2] и легко решающую эту проблему

```
' Процедура "Два шага"
Const N = 2 ' Функция двумерная
Dim X(0 To N)
Sub Кнопка1_Щелкнуть()
    X(1) = [B4]: X(2) = [B5] ' Начальная точка
    Ymin = ДвуЭксп(X(1); X(2)) ' Допущение
    D = 1: DMin = ,0001: J = 0
    Do While D > Dmin ' Цикл уменьшения шага
        Do ' Цикл приближения к минимуму
            P = 0
            For I = 1 To N ' Перебор координат
                For X0 = -D To D Step 2 * D 'Два шага
                    X(I) = X(I) + X0: Y = ДвуЭксп(X(1); X(2))
                    If Y < Ymin Then J = I: P = X0: Ymin = Y
                    X(I) = X(I) - X0
                Next
            Next
            X(J) = X(J) + P ' Переход к новой точке
        Loop Until P = 0
        D = D / 2 ' Уменьшаем шаг в два раза
    Loop
    [B4] = X(1): [B5] = X(2) ' Найденный минимум
End Sub
```

Эта процедура реагирует на нажатие кнопки, расположенной в таблице на рис. 7. Она выдает правильный ответ по всем трем пробным функциям из любой начальной точки, хотя в нее заложен простейший алгоритм. Решатель пакета Excel (как, кстати, программы MathCAD и Eureka [3, 4]) часто попадают в ловушки — в точки максимумов и перегибов.

Пример 5. Решение системы ▲линейных алгебраических уравнений

Есть особый род **извращенцев...** Автору не хватило духа вынести этот термин в заголовок статьи и он его заменил нейтральным — *derivatio*, отклонение. Читатель, может быть, до этого места статьи и не доберется, а об авторе подумает черт знает что. Так вот, есть особый род **дериваторов**, получающих удовольствие от попыток решения на компьютере задач программными средствами, для этих целей прямо не предназначенными. *"А вы ноктюрн сыграть смогли бы на флейте водосточных труб!? А вы корни системы алгебраических уравнений найти смогли бы в среде Решателя пакета Excel!?"*

По идее (и здесь нет никакого, пардон, извращения) поиск корня системы линейных алгебраических уравнений должен быть базовой задачей любого Решателя, так как многие задачи вычислительной математики так или иначе упираются в эту рутинную проблему. Попробуем-ка решить в среде Excel классическую задачу из рассказа А.П.Чехова "Репетитор".

"Купец купил 138 арш. черного и синего сукна за 540 руб. Спрашивается, сколько аршин купил он того и другого, если синее стоило 5 руб. за аршин, а черное 3 руб.?"

На рис. 8 помещена электронная таблица с решением задачи о купце и сукне через команду *Поиск решения...* . Сама система уравнений записана в одноименном диалоговом окне (см. рис. 2) в виде двух ограничений:

B6=B4*B9+B5*B10

B7=B9+B10

В качестве изменяемых ячеек были указаны ячейки B9:B10, где хранятся неизвестные системы — аршинаж, если так можно выразиться, синего и черного сукна. В задаче о купце и сукне целевой функции нет, но если диалоговое окно "Поиск решения" вызывается впервые, то все его поля будут пустыми, требующими заполнений. Все, кроме одного — самого первого — "Установить целевую ячейку", в котором будут записаны координаты активной ячейки таблицы, где находился курсор. Решатель считает, что последняя информация, введенная пользователем в таблицу, — это целевая функция. Поэтому-то в диалоговом окне "Поиск решения" в поле "Установить целевую ячейку" автоматически выставляется соответствующая ссылка. Ни в документации, ни в Help'е пакета Excel нет ни слова о том, что технология "Что-если" может реализовываться без целевой функции. На эту удочку и попался автор, введя в ячейку B12 фиктивную целевую функцию

	Пример 5			
	Задача Удодова			
Стоимость синего сукна	*5*	руб./арш.		
Стоимость черного сукна	*3*	руб./арш.		
Всего потратил денег	*540*	руб.		
Всего купил сукна	*138*	арш.		
	Ответ			
Купил синего сукна	63	арш.		
Купил черного сукна	75	арш.		
	Целевой функции нет			
	Ограничения		Решить систему	
	B7=B9+B10			
	B6=B4*B9+B5*B10			

Рис. 8

B9*B10*1E-100,

которая на всем разумном диапазоне изменения значений ячеек B9 и B10 удовлетворяет условиям точности, если переключатель в окне *"Поиск решения"* поставлен на позицию *"Минимальное значение"*. После такого обманного финта Решатель "отстанет" от целевой функции-пустышки и займется ограничениями — решением задачи, с которой он быстро справится: до отдачи команды *Найти решение...* ячейки B9 и B10 на рис. 8 были пустыми.

Прием подсовывания компьютеру "пустышки" (прием, лишний раз радующий пользователя, напоминая ему, что машина — это машина, а человек — это человек, царь не только живой, но и неживой природы) в ходу не только при работе с Excel, но и при использовании другого популярного Решателя — пакета MathCAD фирмы MathSoft [4]. Нередко в решаемых системах неизвестных оказывается больше, чем уравнений. Человек знает, что это не ошибка, но пакет MathCAD тем не менее настойчиво требует, чтобы число неизвестных не превышало числа уравнений и неравенств (ограничений), зажатых между ключевым словом *Given* и функцией *Find* (*Minerr*). В качестве "пустышки" здесь подойдет какое-нибудь тривиальное тождество, например 1=1. Такая же повышенная "приставучесть" встречается и у языков Pascal и C, где цикл с выходом из середины через функции *break* или *continue* реализуется только обманным путем [5, 6] — записью в заголовке цикла с предпроверкой все той же тривиальной истины 1=1:

```
while 1=1 do begin
       ...
   if... then break; (continue)
       ...
end;
```

Впервые безметочная конструкция "цикл с выходом из середины" была введена в язык Basic *(Do... If... Then Exit Do... Loop)*, а затем перекочевала в язык Pascal (версия Borland Pascal 7.0 — см. выше). Казалось бы, циклы дальше развивать некуда и незачем. Но нет! В языке Basic, интегрированном в Excel 5.0, появился новый вид цикла ***For Each** параметр **In** множество... **Next***, заставляющий по-новому классифицировать циклы в программировании, опираясь на особенности дуэта "пользователь-компьютер".

Цикл 1. Ни компьютер (система программирования), ни человек не знают, когда и где цикл должен быть прерван. Эта конструкция реализуется на языке Basic двумя ключевыми словами ***Do*** и ***Loop***, между которыми зажато тело цикла.

Цикл 2. Компьютер не знает, а человек знает, когда и где цикл должен быть прерван. В этом случае в тело цикла 1 программистом вкрапливаются ключевые слова ***Exit Do*** (Basic) или ***break-continue*** (Pascal и C).

Цикл 3. И человек, и машина знают, сколько раз нужно выполнить циклические действия. В этом случае тело цикла обрамляется ключевыми словами ***For... To... [Step]*** и ***Next*** (Basic). Возможен гибрид циклов второго и третьего типа за счет ввода в цикл с параметром ключевых слов ***Exit For***, обеспечивающих условное или безусловное прерывание цикла.

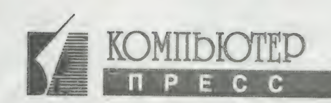
Как понимает читатель, логическая таблица "знает-не знает", положенная в основу вышеизложенной классификации циклов, до сих пор была не полной. Язык Basic пакета Excel 5.0 восполняет эту брешь новым циклом ***For Each** параметр **In** множество... **Next***, предназначенным для ситуаций, когда компьютер знает, сколько раз нужно повторить какие-то действия, а человек не знает или не хочет утруждать себя знанием такой чепухи. Пример: необходимо закрыть открытые электронные книги и открыть закрытые. Компьютер знает, что у него открыто, а что закрыто, человеку же пересчитывать все это не с руки. Вот тут-то и пригодится цикл ***For Each** параметр **In** множество... **Next***.

Цикл первого типа нереален, исходя и из практических, и из эстетических соображений. Во-первых, нет ничего вечного под луной, тем более вечных циклов, а во-вторых, число три (три вида циклов) изящней числа четыре. Автору уже приходилось отмечать [7], что в основу цифровой вычислительной техники положены числа 3 и 7, а не числа 2 (бит) и 8 (байт).

Забегая вперед — к примерам 6 и 7, отметим, что и там описанный прием отказа от целевой функции будет оправдан, если необходимо подобрать начальные параметры оптимизации — опорную точку поиска. Ведь часто бывает так, что Решатель не в силах одновременно и оптимизировать решение, и выполнить ограничения. В этом случае задача решается поэтапно: сначала вводятся все новые и новые ограничения, а в конце вставляется целевая функция. Но часто и до ввода целевой функции дело не доходит: пользователь удовлетворяется тем, что выполнены ограничения, руководствуясь мудрой пословицей: *"Не до жиру, быть бы живу"*. Целевую функцию-пустышку вводить, конечно, не надо. Достаточно оставить в диалоговом окне "Поиск решения" поле "Установить целевую ячейку" **пустым.**

Помещаем ниже Basic-процедуру поиска корней системы *N* линейных алгебраических уравнений, коэффициенты которых записаны в матрице (в двумерном массиве) *A*, а корень записывается в вектор (в одномерный массив) *X*.

```
'Решение системы линейных алгебраических уравнений
Const N = 2 ' Число уравнений (неизвестных)
Dim A(1 To N; 1 To N + 1); X(1 To N)
Sub Кнопка1_Щелкнуть()
   ' Заполнение массивов
   A(1; 1) = [B4]: A(1; 2) = [B5]: A(1; 3) = [B6]
   A(2; 1) = 1: A(2; 2) = 1: A(2; 3) = [B7]
   ' Решение системы
   For J = 1 To N - 1
      For I = J + 1 To N
         A(I; J) = -A(I; J) / A(J; J)
            For K = J + 1 To N
```




```
                A(I; K) = A(I; K) + A(I; J) * A(J; K)
            Next
            A(I; N + 1) = A(I; N + 1) + A(I; J) * A(J; N + 1)
        Next
    Next
    X(N) = A(N; N + 1) / A(N; N)
    For J = N - 1 To 1 Step -1
        P = A(J; N + 1)
        For I = J + 1 To N
            P = P - X(I) * A(J; I)
        Next
        X(J) = P / A(J; J)
    Next
    [B9] = X(1): [B10] = X(2) ' Ответ
End Sub
```

Вторая причина, заставившая автора поместить здесь эту классическую программу — это желание показать непоследовательность развития языка Basic и незавершенность логической схемы циклов, описанной выше. В программе решения системы алгебраических уравнений должны быть задействованы циклы ***For Each** параметр **In** множество... **Next*** вместо циклов ***For... To... Step... Next***. Но! Компьютер знает, сколько раз выполнить цикл, но не знает, как его выполнить, с какого конца. Кроме того, на языке Basic нет перечисляемого типа данных. Помимо этого... Тут автор немного запутался в мыслях. Поэтому оставим "философские" рассуждения и перейдем к новой задаче.

▲Пример 6. Транспортная задача

Автора заставила взяться за эту оптимизационную задачу одна легковесная фраза на стр. 688 руководства пользователя Excel: *"... но очевидно, что перевозить товары на близкие расстояния дешевле"*. Если б это было так, то никакой проблемы под названием "Транспортная задача" в вычислительной математике не было бы, а расчет сводился бы к примитиву: сначала отвезем все, что допустимо ограничениями, по самому дешевому маршруту, затем по маршруту, второму по дешевизне, и т.д. Но не будем голословными и проиллюстрируем парадокс и сложность транспортной задачи конкретным примером.

В городе есть два склада телевизоров и два магазина. Необходимо ежедневно с первого склада вывозить 50 телевизоров, а со второго — 70. Первый магазин при этом получает 40 телевизоров, а второй — 80. Спрашивается, как нужно организовать работу транспорта, чтобы затраты на перевозку были минимальными.

Таблица с решением этой задачи с помощью Решателя пакета Excel показана на рис. 9. И что удивительно, по самому дешевому маршруту (со второго склада в первый магазин — 800 руб./шт.) ничего не надо перевозить, если необходимо добиться минимальных суммарных затрат (134 000 руб./сут.). Этим парадоксом может воспользоваться хозяин транспортного предприятия. Для этого нужно в окне *"Поиск решения"* (см. рис. 3) соответствующий переключатель переставить с позиции *"минимальное значение"* на позицию *"максимальное значение"* и отдать команду *"Выполнить"*. Но Решатель, как бы чувствуя нечистую игру, подведет хитреца: с точки минимума план перевозок на точку максимума сдвинуть невозможно. Решение как бы сидит в некой локальной потенциальной яме, обусловленной принятой методикой оптимизации и точностью вычислений. Попытки автора варьировать параметрами поиска в соответствующем диалоговом окне (см. рис. 4) не привели к успеху. Но это еще опять же полбеды. Настоящая беда в том, что Решатель, как и в задачах 1-3, оставляя решение в точке минимума, по-прежнему нахально врет в глаза: *"Решение найдено, все ограничения выполнены"*.

Примеры данной статьи — это тесты, то есть задачи, решение которых заранее известно. Но в реальной жизни все обстоит гораздо сложней. *"Mendax*

Пример 6						
Транспортная задача						
	План	**Факт**				
Вывозится со склада 1	*50*	50	**тел./сут.**		**Ограничения**	
Вывозится со склада 2	*70*	70	**тел./сут.**		**B4=B15+B16**	
					B5=B17+B18	
Поступает в магазин 1	*40*	40	**тел./сут.**		**B7=B15+B17**	
Поступает в магазин 2	*80*	80	**тел./сут.**		**B8=B16+B18**	
					B15:B18=целое	
	Стоимость перевозок (руб./тел.)				**B15:B18>=0**	
	магазин 1	**магазин 2**				
склад 1	*1200*	*1600*				
склад 2	*800*	*1000*				
Со склада 1 в магазин 1	40	**тел./сут.**	**Суммарная стоимость перевозок (руб./сут.)**			
Со склада 1 в магазин 2	10	**тел./сут.**		134000		
Со склада 2 в магазин 1	0	**тел./сут.**	**В ячейке E16 записана целевая функция**			
Со склада 2 в магазин 2	70	**тел./сут.**	**=B12*B15+C12*B16+B13*B17+C13*B18**			

Рис. 9

in unom, mendax in omnibus — солгавший в одном, лжет во всем", — так утверждает римское право. Люди смягчили этот закон и говорят: *"Единожды солгавший, кто поверит тебе?!"* Но верить приходится, принимая во внимание, что абсолютной истины, к сожалению, а может быть к счастью, нет: у компьютера и у человека разные понятия о том, что такое решение задачи. Это мы уже отметили на элементарном примере 1 поиска корня алгебраического уравнения. В примере 6 "количество" этого противоречия перешло в "качество". Для Решателя, по-видимому, нет большой разницы между минимумом (134 000 руб./сут.) и максимумом (142 000 руб./сут.). Второе число (максимум) можно получить (это подсказка хозяину транспортного предприятия), пойдя на двойную хитрость — выставив перед решением задачи не только переключатель на позицию "Максимальное значение" (см. рис. 3), но и взяв за основу другую опорную точку в ячейках B15:B18 — не нули, а сотни (явно завышенный план). От нулей в точку максимума затрат на перевозку можно также попасть, если не делать остановки на точке минимума.

При решении транспортной задачи (см. рис. 9) стартовая опорная точка (план перевозок) была равна 0,0,0 и 0 т/сут. Но это не установка автора, а умолчание пакета Excel: если ячейки пустые (а так и было в начале решения примера 6 — ячейки B15: B18 были пустые), то Решатель считает, что в них хранятся нули. А это не просто ошибка, а методическая ошибка — при пустых изменяемых ячейках Решатель должен отказываться выполнять задание, требуя от пользователя каких-то начальных установок. Пусть это будут те же нули, но пусть пользователь задумается над тем, "от какой печки нужно начинать плясать", чтобы оптимальным образом решить задачу.

Популярный "школьный" Решатель — математический пакет Eureka фирмы Borland [3] по умолчанию начинает "плясать" не от нуля, а от единицы, что тоже нередко приводит к курьезам и сбоям. "Вузовский" Решатель MathCAD [3, 4] никаких умолчаний не допускает: начальное приближение будет лишним только тогда, когда заранее известно, что решение одно, когда, например, матричным способом решается система линейных алгебраических уравнений.

Кстати, сам пакет Excel далеко не всегда считает, что пустая ячейка — это ячейка с нулем. Иначе функции обработки списков (СУММ, MIN, MAX и др.) работали бы неправильно — не игнорировали бы пустые ячейки, считывая из них нули.

Транспортную задачу реальную, а не учебную, как в примере 6, рекомендуется решать в несколько этапов. Вначале без целевой функции необходимо сделать "пристрелки" — прощупать область существования решений, а уж потом оптимизировать задачу, взяв за опорную точку какой-либо план перевозок, удовлетворяющий ограничениям и расположенный недалеко от оптимума. Ограничения B15:B18=целое, кстати говоря, в задаче 6 лишние, так как экстремальные планы перевозок (134 000 и 142 000 руб./сут.) сами по себе целочисленны, а переход от минимума к максимуму проходит без эксцессов. Отсюда напрашивается вывод: ограничения на целочисленность следует вводить в задачу одним из последних.

Кстати говоря, совсем уж зарвавшийся хозяин транспортного предприятия может химичить не только с целевой функцией, но и с ограничениями типа B15:B18>=0, снимая их и организуя тем самым встречные перевозки.

Пример 7. Задача целочисленного ▲линейного программирования

В оптимизационных задачах приходится сталкиваться не только с дилеммой "Правда-Ложь" (см. выше), но и с дилеммой "Правда-Полуправда", которую (дилемму) мы рассмотрим на примере поиска оптимального плана выпуска компьютеров.

Задачу линейного программирования почти всегда разбирают не в чистом математическом виде, а облекая ее в форму конкретной производственной задачи. Так получается интересней и нагляднее: *"Служил Гаврила почтальоном...", "Служил Гаврила хлебопеком..."* и т.д., если вспомнить сюжет Ильфа и Петрова. Автор помещает эту статью в журнал КомпьютерПресс, поэтому "Гаврила" — это задача линейного программирования, а его текущая специальность — это специальность аудитории, каковой о линейном программировании рассказывают.

Итак (см. рис. 10), необходимо запланировать выпуск компьютеров четырех типов с известной ценой каждого (см. ячейки G12:G15) и известным расходом комплектующих (ячейки B12:F15). В запасе есть определенный набор микросхем пяти типов (ячейки B4:B8). Ситуация отнюдь не вымышленная. Многие компьютерные фирмы часто ограничивают свое производство, ожидая от поставщиков (от фирмы Intel, например) комплектующих, чтобы можно было написать на компьютере "Intel Inside". Это выражается системой семи ограничений:

B4 >= B12*B18+B13*B19+B14*B20+B15*B21
B5 >= C12*B18+C13*B19+C14*B20+C15*B21
B6 >= D12*B18+D13*B19+D14*B20+D15*B21
B7 >= E12*B18+E13*B19+E14*B20+E15*B21
B8 >= F12*B18+F13*B19+F14*B20+F15*B21
B18:B21 >= 0
B18:B21 = целое

Учитывая норов Решателя, проявленный в транспортной задаче, когда он зациклился на минимуме, возьмем в качестве опорной точки заведомо за-

	Пример 7					
Задача линейного целочисленного программирования						
	Есть	Потрачено				
Чип 1	*100*	100	шт			
Чип 2	*20*	20	шт			
Чип 3	*500*	500	шт			
Чип 4	*100*	100	шт			
Чип 5	*1200*	970	шт			
Расход чип (шт./комп.) и стоимость компьютеров						
	Чип 1	Чип 2	Чип 3	Чип 4	Чип 5	Стоимость
Компьютер 1	*1*	*0*	*4*	*0*	*3*	*600*
Компьютер 2	*1*	*0*	*8*	*0*	*8*	*2500*
Компьютер 3	*0*	*1*	*0*	*4*	*15*	*25000*
Компьютер 4	*0*	*1*	*0*	*8*	*64*	*60000*
План выпуска						
Компьютер 1	75	Целевые функции				
Компьютер 2	25		120	Число компьютеров		
Компьютер 3	15		782500	Стоимость компьютеров		
Компьютер 4	5		СУММ(B18:B21) - формула в ячейке D19			
	B18*G12+B19*G13+B20*G14+B21*G15 - формула в ячейке D20					
	Протокол решения					
	Начало	Оптимизация по числу	Оптимизация по стоимости	Оптимизация по числу		
Компьютер 1	100	100	1	75		
Компьютер 2	100	0	56	25		
Компьютер 3	100	20	3	15		
Компьютер 4	100	0	11	5		
Общее число	400	120	71	120		
Стоимость	8810000	560000	875600	782500		

Рис. 10

Ранг, выстраивающий запланированные пользователем действия в некую очередь, используется в среде Excel, например, для сортировки записей в базе данных. Глубина сортировки ограничена тремя полями: сначала записи сортируются по полю, которому пользователь дал высший ранг, потом по полю со вторым рангом, если оно отмечено, а потом уже по третьему полю. Если пользователю трех полей сортировки мало (что встречается довольно редко), то можно перейти на ручной режим глубокой сортировки: сначала отсортировываются записи по полю с наименьшим рангом, потом по полю с предпоследним по значимости рангом и т.д. до поля с высшим рангом сортировки. Изъян такой технологии известен: если необходимо увеличить глубину сортировки еще на одно поле, то всю работу приходится начинать сначала.

вышенный план выпуска компьютеров — 100, 100, 100 и 100 шт. После этого Решатель легко находит оптимальный план (100, 0, 20 и 0 шт.), максимизирующий общее число компьютеров (120). Но это будет уже не Ложью, но еще и не Правдой, а **Полуправдой**: более глубокий анализ задачи 7 показывает, что те же 120 компьютеров можно выпускать сотнями других вариантов. Какой из них самый оптимальный?

Разбирая задачу о купце и сукне (система линейных алгебраических уравнений), мы отметили, что разработчики Решателя вводят пользователей в заблуждение, уверяя его в том, что целевая функция имеет статус обязательного параметра. Число целевых функций может и должно быть не только меньше, но и **больше единицы**. А чтобы Решатель в них не запутался, целевым функциям нужно дать ранг: сначала оптимизируется первая, затем вторая, потом третья и т.д. Но трех, по-видимому, будет достаточно, и не только потому, что три — красивое число [7].

Описанный способ поэтапной (поранговой) оптимизации плана выпуска компьютеров зафиксирован в таблице на рис. 10. Сначала ведется максимизация общей стоимости компьютеров (ячейки D25:D28), затем полученный результат используется в качестве опорной точки новой оптимизации (ячейки E25:E28) по новой целевой функции (число компьютеров) — по функции первого ранга. Так и был получен оптимальный план выпуска компьютеров (120 штук) с более высокой общей стоимостью (с более оптимальным использованием имеющихся ресурсов).

Кстати говоря, ранг важности было бы неплохо присваивать и ограничениям, накладываемым на задачу. Ведь в реальной жизни одни ограничения имеют статус табу, нарушить которое недопустимо ни при каких обстоятельствах, другие же можно рассматривать как пожелания. Решатель же пакета Excel имеет свои собственные представления о важности ограничений, и при анализе не полностью решенной задачи нередко оказывается так, что какое-

нибудь пустяковое ограничение выполнено, а самое важное — нет. Ограничения несложно учесть и при реализации метода "Два шага" (см. программу выше) вводом штрафных санкций.

Задачу, сформулированную в таблице на рис. 10, можно решить не ранговой оптимизацией, а вводом единственной, но "хитрой" целевой функции — произведения числа компьютеров на общую их стоимость. Но к этому способу решения как бы душа не лежит. Термин "целевая функция" идет от слова "цель", а она (цель) должна иметь четкий физический смысл, чего нет в "хитром" произведении общей суммы на общую стоимость.

Понятие "физический смысл", только что затронутое нами, наталкивает на идею совершенствования механизма электронных таблиц, связанную с вводом в ее ячейки не только значений величин, но и их **размерностей** (метры, килограммы, секунды и т.д.). Эта идея уже реализована в упоминавшихся пакетах MathCAD и Eureka, где можно дополнить правильность ввода формул проверкой соответствия размерностей: метры невозможно сложить с килограммами [3, 4]. В среде Excel и в средах других электронных таблиц зачатки размерности есть, но они применяются только при выводе чисел в денежном формате, что объясняется финансовой (в первую очередь) направленностью расчетов. Таблицы на рис. 8, 9 и 10 содержат величины с размерностью (руб., арш., руб./арш., шт./сут., руб./шт. и т.д.), но проверка в формулах ограничивается лишь тем, чтобы число не складывалось с текстом и не более того.

Кстати, общую стоимость компьютеров (см. ячейку D30 на рис. 10) можно еще существенно увеличить, сняв с задачи ограничение D2>=0. Тогда появятся отрицательные значения числа компьютеров первого типа. Это можно рассматривать и как ошибку, и как еще одну "хитрую" подсказку пользователю: "Купи на стороне дешевые компьютеры первого типа, вытащи из них дефицитные микросхемы и пусти их на производство более дорогих ЭВМ. План по валу в рублях будет перевыполнен!"

▲Выводы

Собака после выстрела охотника и команды "Искать!" может вернуться с тремя результатами:

1. Собака приносит подстреленную дичь.
2. Собака, израсходовав отпущенный лимит времени или услышав призывный сигнал охотника, возвращается с пустыми зубами.
3. Охотник получает "трофей" — стреляный пыж или рваный башмак, но собачьи глаза и хвост выдают собачью хитрость.

Есть особый род **эстетов**, получающих удовольствие от попыток облагораживания технического текста цитатами из художественной литературы. Но не этот смысл слова "эстет" подчеркнут в названии первой части статьи. Автору уже доводилось признаваться своим читателям [6], что для него в программной среде огромную роль играют не только чисто технические параметры (быстродействие, богатство встроенных функций и т.д.), но и "красота" продукта. Под красотой же следует понимать не только цветовую палитру, например, или изящество окон и иконок, но и то наслаждение, какое программа может доставлять при решении даже простейших задач. И не только своими новыми, ранее неведомыми возможностями, но и своими... ошибками. Автор попытался было описать всю эту гамму чувств, но его опередил Карел Чапек своим рассказом "Игла", где необыкновенно точно передан дух работы с таблицами — и не важно, бумажные они или электронные. Помимо этого, чапековский отрывок даст понять, причем тут охотник.

«Когда я еще работал в бухгалтерии и составлял, бывало, полугодовой отчет, подчас случалось, что цифры не сходятся. Однажды в наличности не хватило трех геллеров. Конечно, я мог просто положить в кассу эти три геллера, но это была бы неправильная игра. С бухгалтерской точки зрения это было бы неспортивно. Надо найти, в каком счете допущена ошибка, а счетов у нас было четырнадцать тысяч. И скажу вам, когда я брался за баланс, мне всегда хотелось, чтобы там обнаружилась какая-нибудь ошибка. Тогда я, бывало, оставался на службе хоть на всю ночь. Положу перед собой кучу бухгалтерских книг и берусь за дело. И для меня колонки цифр становились не цифрами, они преображались просто необыкновенно. То мне казалось, что я карабкаюсь по этим колонкам вверх, словно на крутую скалу, то я спускаюсь по ним, как по лестнице, в глубокую шахту. Иногда я чувствовал себя охотником, который продирается сквозь чащу цифр, чтобы изловить пугливого и редкого зверя — эти самые три геллера. Я воображал себя охотником, который бродит по горам, по долам, среди росистых кустиков черники. И до того мне в такие минуты становилось хорошо от этого ощущения движения и силы, такое я чувствовал вокруг себя волнующее приволье, словно и в самом деле переживал необыкновенное приключение. Целыми ночами я мог охотиться за тремя геллерами, и когда находил их, то даже не думал, что это лишь жалкие гроши. Это была добыча, и я шел спать, торжествующий и счастливый...»

Работа с Решателем пакета Excel во многом напоминает ружейную охоту. Пользователь формирует таблицу, пишет целевую функцию, вводит ограничения, выставляет параметры поиска (заряжает ружье, выслеживает дичь), а затем щелкает по кнопке

"Выполнить" в окне "Поиск решения" (нажимает спусковой крючок ружья). Собака (Решатель) прыгает в воду и возвращается с тремя результатами:

1. Решение найдено.
2. Решение не найдено.
3. Решатель пытается подсунуть пользователю что-то такое, что только издали напоминает правильный результат.

При этом ситуация осложняется тем, что пользователь пакета Excel в отличие от охотника, который знает, что такое дичь, часто не знает, как выглядит правильный результат. У Решателя, к сожалению, нет глаз и хвоста, по которым можно определить, что принесено к ногам охотника — пользователя Excel.

Оптимизация в среде Решателя Excel — это хождение на охоту с чужой собакой, способности и повадки которой неизвестны. "Своя собака" — это методы и алгоритмы, написанные на языке Basic и вставленные в модуль рабочей книги, по которым решение ищется не вслепую, а наверняка. А если писать программы недосуг, то вот семь советов по работе с Решателем Excel:

1. Найдя решение, еще раз заставьте Решатель найти его уже от новой опорной точки.
2. Перед решением оптимизационной задачи с ограничениями прогоните ее без целевой функции для того, чтобы оценить область существования решений.
3. Начинайте поиск оптимального решения от одной из точек, найденной в п.2 и лежащей недалеко от оптимума.
4. Вводите ограничения постепенно: ввели первое — нашли решение, ввели второе — уточнили решение и т.д.
5. Начинайте решение оптимизационной задачи с целочисленными аргументами без ограничений на целочисленность. Нецелочисленный ответ всегда будет полезен, так как (1) от него можно будет начать поиск целочисленного решения, (2) можно удовлетвориться нецелочисленным решением (появление совместителей, например, в задаче об оптимальном штатном расписании) и (3) числа можно округлить вручную, если целочисленное решение не находится.
6. Проверяйте правильность найденных минимумов и максимумов построением графиков и поверхностей, где эти особые точки видны. Благо пакет Excel имеет богатый набор деловой графики. Если аргументов у функции больше двух, то постройте графики сечений через точку оптимума по всем координатам и убедитесь, что там все частные производные равны нулю.
7. Имейте набор Basic-процедур и функций для решения конкретных оптимизационных задач не лобовой атакой через использование Решателя, а более осмысленными способами. Три подобные программы приведены в статье.

Литература:

1. Банди Б. Методы оптимизации. Вводный курс. Пер. с англ. М.: Радио и связь, 1988
2. Очков В., Пухначев Ю. 128 советов начинающему программисту. 2-е издание. М.: Энергоатомиздат, 1992
3. Очков В., Пухначев Ю. Уроки для пользователей IBM PC. М.: Финансы и статистика, 1992
4. Очков В. Сказ про то, как MathCAD задачу решал. КомпьютерПресс, № 1-2, 1995
5. Очков В. Turbo Pascal 7.0. Взгляд со стороны. КомпьютерПресс, № 7, 1993
6. Очков А., Очков В. Visual Basic и формула Вирта. КомпьютерПресс, № 7-8, 1994
7. Очков В. Рахаев М. Этюды на языках QBasic, QuickBasic и Basic Compiler. М.: Финансы и статистика, 1995

Рассказ о возможностях пакета Picture Man фирмы СТОИК — мощного средства работы с графическими и видеоизображениями.

Picture Man — новая степень свободы

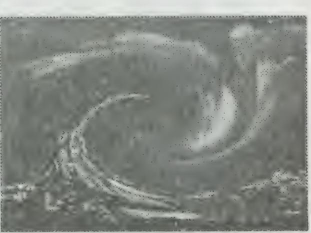
Ольга Ускова,
Василий Логинов

Нам ежедневно приходится работать с различными видами информации. С текстовой информацией мы успешно справляемся благодаря текстовым процессорам и системам оптического распознавания. Но завершенный вид нашим сообщениям и презентациям, деловой переписке и статьям придают профессионально сделанные, эффектные графические изображения. Нескромный вопрос — откуда они берутся?

Для профессиональных дизайнеров, занимающихся, например, разработкой рекламных объявлений, на компьютерном рынке предлагаются мощные и довольно дорогие программные решения — Adobe Photoshop, Aldus PhotoStyler, наконец, знаменитый пакет CorelDRAW!. Средняя цена таких продуктов на российском рынке колеблется от 450 до 900 долларов. Фирмы — разработчики этих продуктов не имеют российских представительств; как правило, эти продукты плохо или вообще никак не локализованы. Грамотную техническую поддержку по этим программам в России получить очень сложно. Полноценно работать с такими инструментами могут только профессионалы, чьи услуги тоже недешевы.







Условимся, что наиболее распространенный путь работы с бумагами в офисе — вообще их не оформлять — для нас непригоден. Текст без графики проигрывает в наглядности и презентабельности, с безлико оформленными материалами трудно претендовать на международный уровень работы.

Сегодня мы хотим представить программные средства серии Picture Man, разработчиком которых является российская фирма СТОИК, а основным продавцом в России и странах СНГ — маркетинговый партнер фирмы СТОИК, компания Cognitive Technologies. «Картинный Человек» выгодно отличается весьма простым и понятным интерфейсом — и притом насквозь русскоязычным. Особенностью интерфейса Picture Man является его перестраиваемость — он основан на комплексе плавающих панелей и окон. Разработка, кстати, изначально базировалась в научно-производственном объединении «АстроФизика».

Для того чтобы оформить письма, отчет, доклад, презентацию, можно воспользоваться рисовалкой, находящейся в программе, а можно применить совокупность средств морфинга и превращений, оригинальных фильтров и динамических спецэффектов, предлагаемых Picture Man. Готовое изображение обычно берется из дискового файла (конвертер Picture Man понимает практически все существующие графические форматы), с бумажного листа (Picture Man работает со сканерами) или видеомагнитофона (поддерживается работа с платами видеоввода). Затем это изображение можно несколькими способами улучшить, а также наложить любой из 80 (или 250, в зависимости от версии) фильтров и создать различные спецэффекты. Так что о нарушении авторских прав можно не беспокоиться — после обработки изображений средствами Picture Man их родная мама не узнает.



Надо сказать, что интерфейс обработок Picture Man открыт, и в систему легко включить любые технологии обработки изображений, например OCR, измерение характеристик изображений или проверку подлинности документов. Уже выпущены система поддержки стандартов пользовательских фильтров Adobe Photoshop Plug-In и модуль OCR фирмы Cognitive Technologies. К моменту публикации этой статьи будут готовы технология Multilayers и библиотека стилизаций реальных изображений под различные художественные манеры (несколько типов стилизаций, например «гравюра», «рисунок пастелью» и другие, включены в стандартную поставку Picture Man).

После того как изображение, которое необходимо включить в документ, готово (устранены дефекты, удалены лишние детали, добавлены недостающие, композиция изменена и все, что можно, деформировано), оно может быть внедрено в документ несколькими способами — как графический файл, содержимое Буфера Обмена Windows или объект OLE 2.0, если это позволяет приложение-клиент, например текстовые процессоры и электронные таблицы пакетов Microsoft Office, Lotus SmartSuite и Novell Office.

Работа с OLE 2.0 предпочтительней не только потому, что позволяет редактировать графический объект средствами Picture Man по месту в приложении-клиенте OLE 2.0. Picture Man 3.0 поддерживает протокол обмена DDE и командный язык. Это позволяет реализовать работу в пакетном режиме в приложениях-клиентах, имеющих собственный командный язык, такой как WordBasic в Microsoft Word 6.0 или Visual Basic в Microsoft Excel 5.0. Таким образом, можно автоматически редактировать большие пакеты изображений, например, с целью устранения типовых дефектов, преобразования графических форматов и так далее.

В состав системы Picture Man 3.0 входит также программа Video Man, созданная на базе технологии Microsoft Video for Windows. Video Man позволяет создавать анимационные файлы формата Microsoft Video (AVI) из последовательностей графических файлов. Анимации также можно внедрять через интерфейс OLE 2.0 в документы, таблицы и презентации. Picture Man поможет, например, подготовить видеоролик с морфологическими преобразованиями рекламируемых товаров, который можно внедрить в презентацию Microsoft PowerPoint или Novell Presentations.

Количеством разнообразных регалий Picture Man не уступает зарубежным конкурентам. Это единственная российская программа, удостоенная титула Windows User Best Buy. До того как маркетингом Picture Man в России и СНГ занялась фирма Cognitive Technologies, продукт продавался в США (фирма Pirus), во Франции (Sybex), в Корее (Darim), в Германии (Wise Software, Egeling Software). Цены для России и СНГ на порядок ниже мировых, маркетинг Picture Man в России проходит под девизом «Современному способу ведения делопроизводства — доступные цены». Горячая линия поддержки пользователей и деловых партнеров работает сразу в двух московских компаниях, СТОИК и Cognitive Technologies.



Windows / C++

Дмитрий Рамодин

Вопрос: Не могли бы вы объяснить, как одновременно использовать в одной программе, написанной с помощью компилятора Borland C++ 4.0, нестандартные элементы библиотек BWCC и CTL3D?

М.Миткевич, г.Хотьково, Московской области

Ответ: Действительно, компиляторы языка C++ фирмы Borland позволяют использовать нестандартные элементы в диалоговых панелях. Чтобы ответить на вопросы о совместимости, необходимо рассмотреть основные принципы работы библиотек нестандартных элементов. Библиотека трехмерных элементов CTL3D фирмы Microsoft подменяет стандартные элементы диалогов типа “BUTTON”, ”EDIT”, “LISTBOX” и т.д. при помощи глобального сабклассинга (subclassing) — особого метода наследования объектов. Он основан на замене оконной процедуры стандартного элемента диалога своей оконной процедурой, которая сама перехватывает и обрабатывает сообщения Windows. Таким образом, библиотека имеет возможность отрисовывать элементы диалога на свой манер. Теперь каждое обращение программы к стандартным диалоговым элементам будет перехвачено и обработано библиотекой CTL3D. Подробнее об этой библиотеке Вы можете узнать из статьи А.Федорова “Библиотека CTL3D — организация и использование” (КомпьютерПресс №6’94 с.10-12).

Библиотека BWCC является противоположностью библиотеки CTL3D. При ее загрузке в память компьютера происходит регистрация новых оконных классов в системе Windows. Значит, чтобы сделать диалоговое окно в стиле BWCC, необходимо:

1. Создать ресурс диалоговой панели в редакторе ресурсов.
2. Явно указать принадлежность элементов управления к классам библиотеки.

В компиляторе Borland C++ версии 4.0 и выше Вы можете вызвать редактор ресурсов Resource Workshop и в ответ на его вопрос о типе создаваемого окна диалога выбрать опцию Borland dialog. Теперь созданный Вами диалог автоматически будет отображаться в стиле BWCC.

К сожалению, в компиляторе Borland C++ нельзя задать одновременное использование библиотек CTL3D и BWCC, но компилятор можно “обмануть”. Если Вы создаете текст при помощи AppExpert, то установите опцию Control Style | 3D для разрешения работы с библиотекой CTL3D, нужные вам параметры и запустите генерацию исходного кода. В полученных исходных текстах найдите определение метода InitMainWindow. В конце него будет стоять вызов функции EnableCtl3d(TRUE). Эта функция загружает и подключает библиотеку CTL3D. Теперь нужно добиться, чтобы другой диалог, созданный вами в стиле, отображался на экране правильно. Для этого добавьте сразу после вызова функции EnableCtl3d(TRUE) вызов функции EnableBWCC(TRUE). Теперь библиотека BWCC тоже будет загружена в память, и ее стиль элементов будет доступен. Пример совместного использования библиотек CTL3D и BWCC показан ниже, а результат его работы отображен на рисунке.

```
//----------------------------------------
//
//    Файл CPTSTAPP.H
//
//----------------------------------------

#if !defined(__cptstapp_h)
#define __cptstapp_h

#include <owl\owlpch.h>
#pragma hdrstop

#include <owl\editfile.h>
#include <owl\opensave.h>

#include "cptstapp.rh"

class cptestApp : public TApplication {

public:
     cptestApp ();
     virtual ~cptestApp ();

public:
     virtual void InitMainWindow();
     void CmDialogs();    // Обработчик команды
                          // File|Dialogs
```

```
 DECLARE_RESPONSE_TABLE(cptestApp);
};

#endif

//-------------------------------------------
//
//    Файл CPTSTAPP.RH
//
//-------------------------------------------
#define IDI_SDIAPPLICATION   1001    //  Иконка программы
#define SDI_MENU             100     //  Меню ID

#define CM_EXIT              24310
#define CM_DIALOGS           24311

//-------------------------------------------
//
//    Файл CPTSTAPP.RC
//
//-------------------------------------------
#if !defined(WORKSHOP_INVOKED)
#include <windows.h>
#endif
#include "cptstapp.rh"

SDI_MENU MENU
{
 POPUP "&File"
 {
  MENUITEM "Dialogs", CM_DIALOGS
  MENUITEM "E&xit\tAlt+F4", CM_EXIT
 }

}

IDI_SDIAPPLICATION ICON "applsdi.ico"

501 DIALOG 35, 31, 194, 77
STYLE DS_MODALFRAME | WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU
CAPTION "CTL3D Dialog"
FONT 8, "MS Sans Serif"
{
 DEFPUSHBUTTON "OK", IDOK, 72, 52, 50, 14
 CTEXT "Hello from CTL3D!", -1, 61, 15, 60, 8
}

502 DIALOG 6, 15, 189, 85
STYLE DS_MODALFRAME | WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU
CLASS "bordlg"
CAPTION "BWCC Dialog"
FONT 8, "MS Sans Serif"
{
 CONTROL "", -1, "BorShade", BSS_HDIP | BSS_LEFT | WS_CHILD |
        WS_VISIBLE, 0, 42, 189, 3
 CONTROL "", IDOK, "BorBtn", BS_DEFPUSHBUTTON | WS_CHILD | WS_VISIBLE
        | WS_TABSTOP, 76, 54, 37, 25
 LTEXT "Hello from BWCC", -1, 64, 14, 60, 8
}

//-------------------------------------------
//
//    Файл CPTSTAPP.CPP
//
//-------------------------------------------

#include <owl\owlpch.h>
#pragma hdrstop
#include "cptstapp.h"

DEFINE_RESPONSE_TABLE1(cptestApp, TApplication)
 EV_COMMAND(CM_DIALOGS,CmDialogs), // Обработать команду
                                   // File|Dialogs

END_RESPONSE_TABLE;
```

```
class  TTestWindow : public TFrameWindow {
public:
 TTestWindow (TWindow *parent, const char far *title, TWindow
        *clientWnd,
 BOOL trackMenuSelection = FALSE, TModule *module = 0) :
 FrameWindow(parent, title, clientWnd,
 trackMenuSelection, module) {}
      ~TTestWindow () {}
};

cptestApp::cptestApp () : TApplication("Computer Press Test")
{
}

cptestApp::~cptestApp ()
{
}

void cptestApp::CmDialogs ()
{
      (new TDialog(MainWindow, 501))->Create(); // Выводим CTL3D-
         диалог
      TDialog(MainWindow, 502).Execute();     // Выводим BWCC-диалог
}

void cptestApp::InitMainWindow ()
{
      TTestWindow *frame = new TTestWindow(0, GetName(), 0, FALSE);

      nCmdShow = nCmdShow != SW_SHOWMINIMIZED ? SW_SHOWNORMAL :
        nCmdShow;

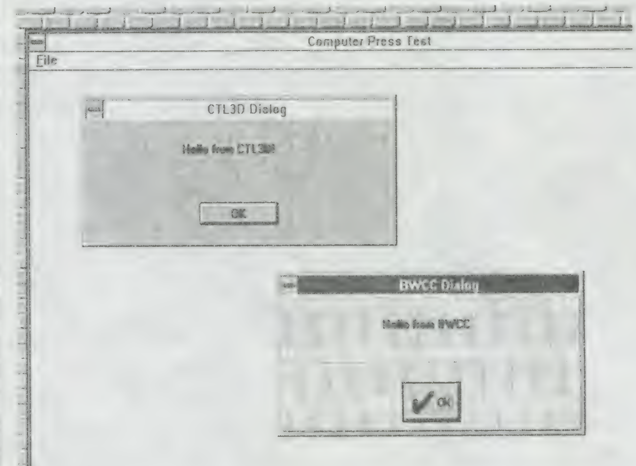
      frame->SetIcon(this, IDI_SDIAPPLICATION);
      frame->AssignMenu(SDI_MENU);
      MainWindow = frame;

      EnableCtl3d(TRUE);  // Загрузить библиотеку CTL3D
      EnableBWCC(TRUE);   // Загрузить библиотеку BWCC
}

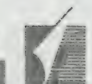
int OwlMain (int , char* [])
{
    cptestApp App;
    int       result;

    result = App.Run():

    return result;
}
```



Совместное использование библиотек CTL3D и BWCC

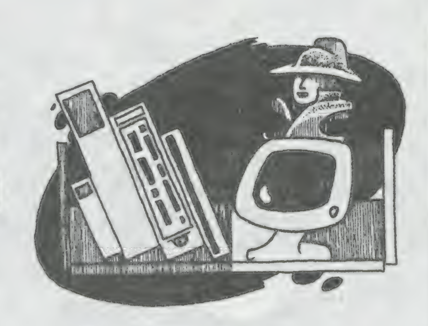


Вопрос: Не могли бы вы объяснить в общих чертах, как можно создавать VBX-элементы при помощи компилятора Borland C++ 4.0?

Ответ: Элементы Visual Basic Extension (VBX) появились одновременно с компилятором Visual Basic фирмы Microsoft и сразу приобрели популярность. Для программистов на C++ или Pascal это очень удобный "черный ящик". Нет никакой необходимости знать, что у него внутри. Встроив VBX типа "таблица" в Вашу программу и написав минимум кода, Вы получаете готовую электронную таблицу с изменяемыми размерами и выделяемыми ячейками. И это за 10 минут! Еще одно немаловажное достоинство VBX в том, что их написано огромное количество. Их выбор покрывает все области разработки программных средств. Хороший пример набора VBX — пакет Visual Solutions Pack фирмы Borland. Производители компиляторов тоже обратили внимание на VBX и включают поддержку работы с ними в свои продукты. К примеру, в составе компилятора Borland C++ 4.0 имеется специальная утилита для извлечения необходимой информации из VBX-элементов под названием VBXGEN, а в библиотеке OWL есть специальный класс TVbxEventHandler, облегчающий подключение VBX к проекту. В версии 4.5 компилятора Borland C++ реализована поддержка VBX-элементов для 32-битных приложений.

Если Вы хотите создавать свои собственные VBX-элементы, то Вам необходимо приобрести либо пакет Visual Basic 3.0 Professional, либо пакет Visual Controls Pack фирмы Microsoft. В них включено средство для создания собственных VBX, которое называется Custom Control Development Kit (CDK). CDK включает в себя необходимые заголовочные (h)-файлы, исходные тексты на языке С, справочную систему и т.д. Используя CDK и любой компилятор языка С, поддерживающий стандарт ANSI C, Вы сможете создавать свои VBX-элементы. Так как компилятор Borland C++ поддерживает этот стандарт, то включив опцию Options|Compiler|Source|ANSI, Вы можете использовать его для компиляции своих собственных VBX. Думая о VBX, будьте осторожны, поскольку фирма Microsoft отказалась от дальнейшего развития VBX-элементов и взяла курс на новые нестандартные элементы — OLE Custom Control Extension (OCX). Они полностью заменят VBX с распространением программных продуктов, поддерживающих протокол OLE 2. Сам OCX представляет собой маленький OLE-сервер, который по запросу программы-клиента, использующей OCX, способен отрисовывать сам себя и реагировать на внешние события. Более подробно мы расскажем об OCX в одном из следующих номеров нашего журнала.

Новые книги издательства «Питер»



Камилл Ахметов

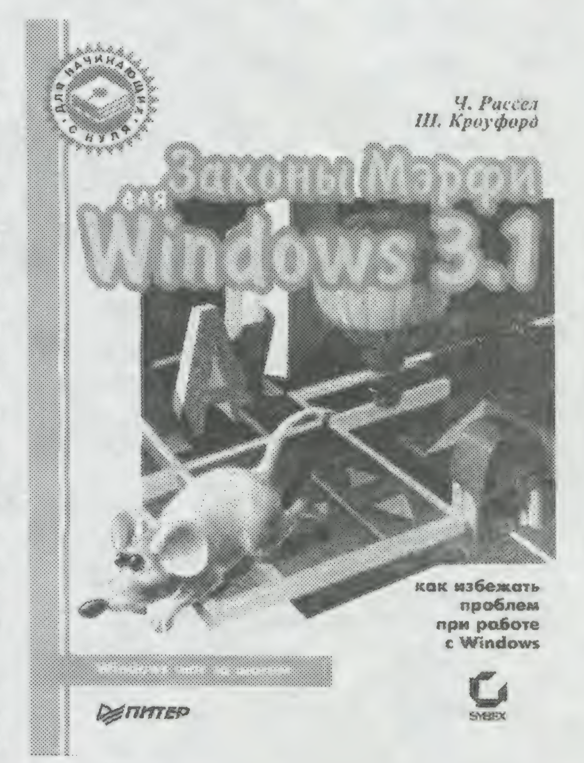
Мы уже представили в нашей «Книжной полке» издательство «Питер» с весьма удачными книгами Б.С.Богумирского. Вскоре после Нового года наши Санкт-Петербургские коллеги представили две книги, лицензированные у американских издательств.

«Законы Мэрфи для Windows 3.1» Ч.Рассела и Ш.Кроуфорда. Серия «Законы Мэрфи» издается фирмой SYBEX Computer Books и конкурирует с серией IDG Books «For Dummies». Соответственно книга «Законы Мэрфи для Windows 3.1» соперничает с «Windows 3.1 для «чайников» Энди Ратбона.

«Законы Мэрфи» имеют следующие отличительные особенности. Несмотря на непременные реверансы типа: «Эта книга — справочник. Вы можете мельком заглянуть в нее, найти ответ на интересующий вас вопрос и тут же ее захлопнуть» и «Вы вовсе не собираетесь становиться компьютерным гуру»,— она все же представляет собой систематическое изложение сведений о Windows в чрезвычайно доступной форме. Там, где это необходимо, появляются практические рекомендации. Если угодно, это «Курс молодого бойца» от SYBEX.

Поэтому изложение в целом выглядит достаточно стройным и логичным. «Законы» Мэрфи (вроде знаменитого «Если какая-нибудь неприятность может случиться, она случается») и различные жизненные утверждения других личностей (кстати, весьма меткие, например: **«Закон Гейтса:** Унция репутации стоит фунта работы») не то чтобы очень нужны, но приятны глазу и разнообразят содержание книги.

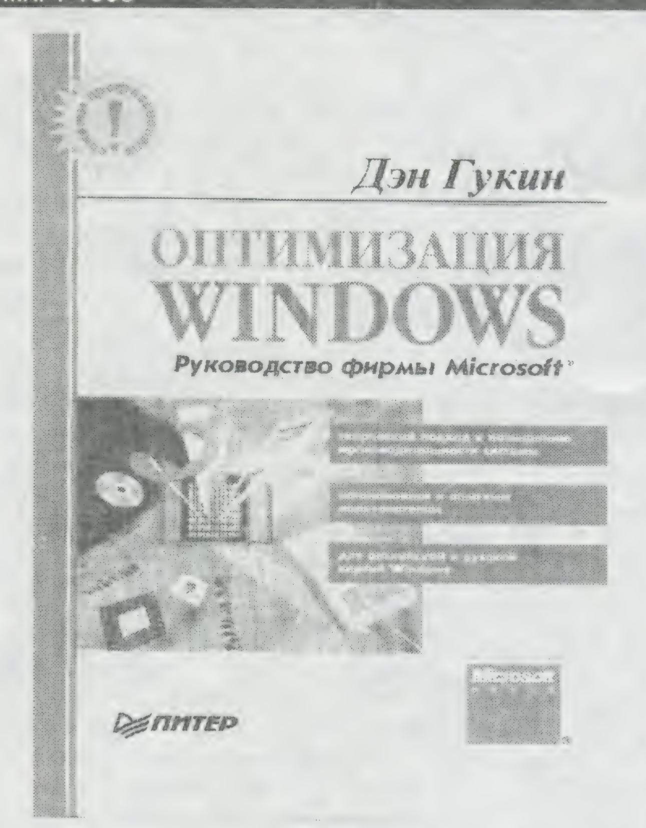
Направленности книги на начинающих пользователей соответствует и то, что в ней достаточно много внимания уделено начальным знаниям по аппаратному обеспечению и системе DOS. Отдельная глава посвящена особенностям Windows для рабочих групп. Подробно описана работа с программой Microsoft Backup из MS-DOS 6.x, что весьма ценно, — может быть, несколькими пользователями, производящими резервное копирование, благодаря этому станет больше. Достаточно много места уделено типичным проблемам, которые возникают при работе Windows-приложений и программ DOS под Windows.



«Оптимизация Windows» Дэна Гукина выпущена издательством «Питер» по лицензии Microsoft Press. «...Windows — это гораздо больше, чем просто удобный графический интерфейс, развитая система меню и работа с мышью, — написал Дэн Гукин во введении к книге. — За наружным фасадом Windows скрыт мощный и сложный двигатель, которому для обеспечения наивысшей производительности необходима правильная настройка. Эта настройка состоит из несложных приемов, таких как конфигурирование дискового кэша, создание постоянного файла подкачки, управление механизмом быстрого переключения между выполняемыми программами и т.п. ...В Windows заложен большой потенциал повышения производительности. Но, как любой сложный и тонкий механизм, Windows требует вдумчивого и умелого обращения».

Книга начинается с описания правильной настройки AUTOEXEC.BAT и CONFIG.SYS для запуска Windows. Затем рассказано об управлении запуском Windows, работе Windows с аппа-





ратурой. Весьма интересны и полезны глава 4, в которой подробно рассказано об оптимизации использования системных ресурсов, памяти и жесткого диска; глава 5, посвященная выполнению приложений DOS из-под Windows; и глава 6, в которой рассказано об оптимизации работы в многозадачном режиме. В той же 6-й главе подробно рассказано о применении методов DDE и OLE. Остаток книги посвящен графическим возможностям Windows (преимущественно работе со шрифтами), использованию Windows и Windows для рабочих групп в сетях и телекоммуникациях, а также мультимедиа-возможностям Windows.

Кроме перечисленных тем в книге освещены особенности версий MS-DOS 6.x и русских версий Windows. Соответствующие разделы добавлены в книгу издательством «Питер».

Остается заключить, что я провел немало интересных часов с этой книгой и системой Windows. Честно говоря, наличие не менее трех-четырех однотипных книг по Windows 3.1 и каждому приложению Microsoft Office уже не так радует глаз, как год назад. Хочется либо очень качественной продукции, такой как книги «Русской Редакции» (Компьютер-Пресс №2'95), либо действительно интересных и практически полезных книг, как «Оптимизация Windows».

Электронные «матрешки» от Texas Instruments

Все познается в сравнении.
Народная мудрость

Сергей Юдицкий

В последние годы в России получили широкое распространение современные технологии телекоммуникации и обработки информации. Это произошло благодаря выходу на российский рынок ряда крупных фирм, занимающихся разработкой и производством такого рода оборудования. Аппаратные средства связи, управления, сбора и обработки информации в настоящее время немыслимы без использования процессоров цифровой обработки сигналов (ЦОС). В мире существует довольно много фирм, занимающихся производством процессоров ЦОС. Но основными производителями по доле занимаемого рынка можно считать следующие фирмы: Motorola, Texas Instruments, Analog Devices, NEC, AT&T. В странах бывшего СССР большую популярность приобрели процессоры американских фирм Texas Instruments и Analog Devices. Отличительной особенностью процессоров этих фирм (что и определило их широкое распространение среди разработчиков) можно считать сравнительно низкую стоимость при высоком быстродействии. Эти две фирмы ведут разработку своих процессоров примерно в одинаковых направлениях, в основе архитектуры процессоров обеих фирм лежит Гарвардская архитектура, и сложно сказать, процессоры какой фирмы лучше. Единственно можно отметить, что при необходимости большого объема математических вычислений (таких как фильтры, в особенности гребенчатые, корреляторы и т.п.) несколько лучше использовать процессоры фирмы Analog Devices, поскольку их внутренняя архитектура лучше приспособлена для этого. При построении же многопроцессорных систем, контроллеров либо систем, где необходим активный обмен информацией с различными внешними устройствами, предпочтительнее процессоры фирмы Texas Instruments, поскольку они имеют большее количество и большую гибкость периферийных устройств. Процессоры ЦОС Analog Devices уже были представлены читателям в предыдущих номерах КомпьютерПресс (№8' 94 и №1' 95). Данная работа является логическим продолжением этой темы, она знакомит вас с увлекательным миром сигнальных процессоров с фиксированной точкой от Texas Instruments (TI).

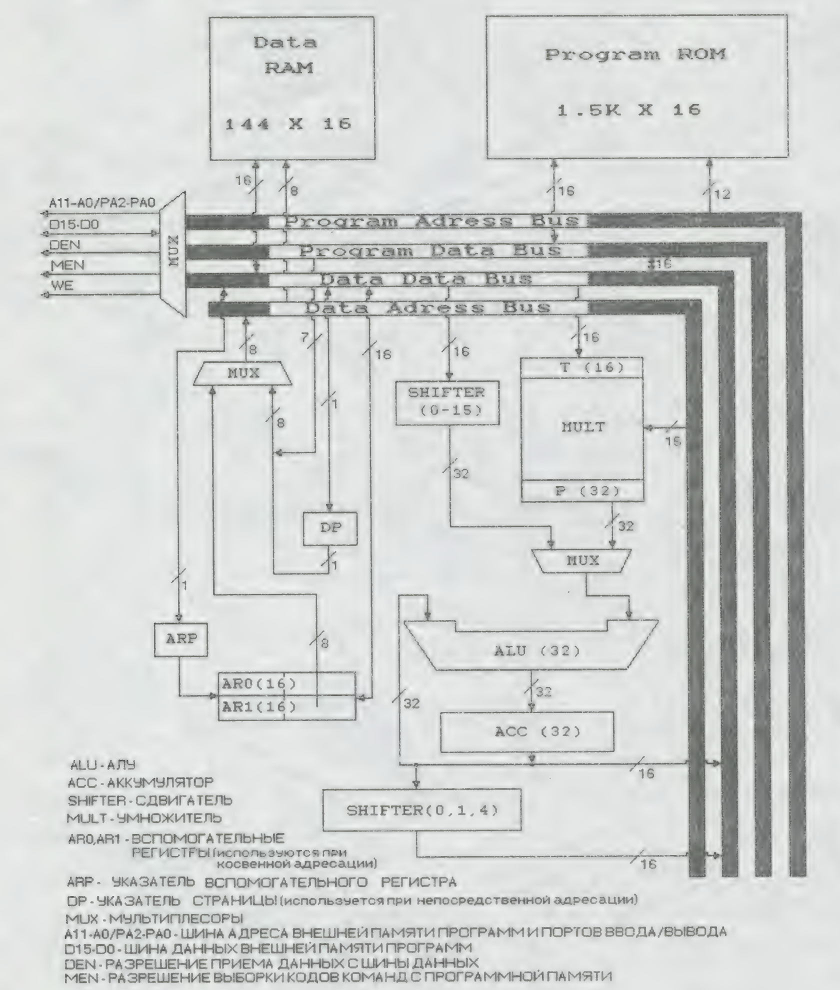


Рис. 1. Структурная схема процессора TMS320C10

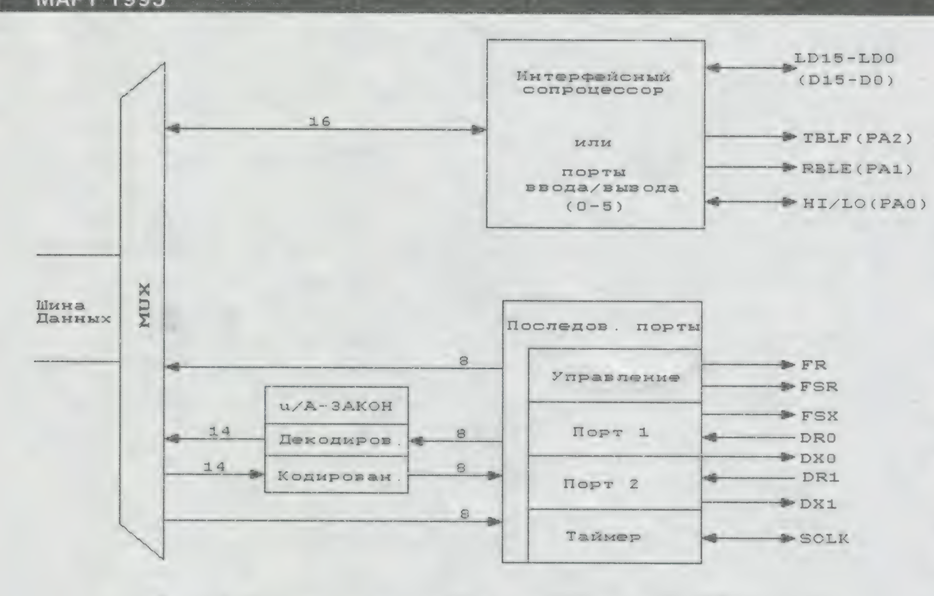


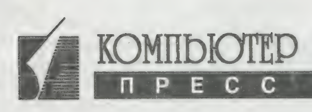
Рис. 2. Структура интерфейса процессора TMS320C17

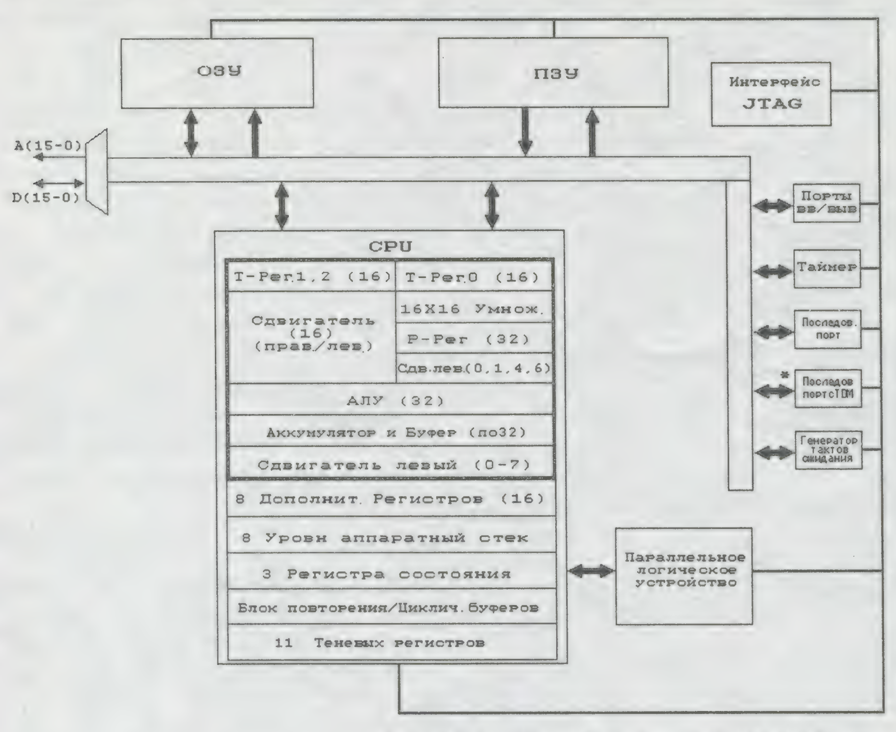
Процессоры ЦОС фирмы TI разделяются на два больших семейства: это процессоры для обработки чисел с фиксированной точкой и процессоры для обработки чисел с плавающей точкой. Семейство процессоров с фиксированной точкой состоит из трех поколений, базовыми моделями которых являются соответственно TMS320.10, .20, .50. Семейство процессоров с плавающей точкой состоит из двух поколений, их базовые модели: TMS320.30 и .40.

Процессоры старших поколений поглощают в себе архитектуру и систему команд младших поколений. При этом семейства несовместимы между собой.

Первый процессор ЦОС TMS320.10 был выпущен фирмой TI в 1982 году и является наиболее дешевым процессором. Его структурная схема представлена на рис. 1. В основу процессора положена модифицированная Гарвардская архитектура. В строго выдержанной Гарвардской архитектуре память программ и данных расположена в различных адресных пространствах, то есть существует четыре основные шины: команд, данных, адреса команд, адреса данных, что позволяет одновременно осуществлять выборку и выполнение команд. Модификация Гарвардской архитектуры обеспечивает обмен данными между памятью программ и памятью данных, что повышает гибкость устройства. Другие модификации процессоров первого поколения ('C14, 'C15, 'C16, 'C17) имеют такую же архитектуру, но различаются конфигурацией памяти, наличием или отсутствием дополнительных периферийных устройств и временем командного цикла. Эти особенности будут описаны далее. Представленная таблица достаточно полно иллюстрирует указанные различия. Процессор оснащен 16-разрядной шиной данных. Арифметические функции в процессоре реализованы аппаратно. Он имеет аппаратный умножитель (MULT), позволяющий за один командный цикл (200 или 160 нс) перемножить два 16-разрядных числа, получив 32-разрядный результат. Аппаратно реализованное устройство сдвига (SHIFTER) дает возможность сдвигать данные во время их пересылки в АЛУ (ALU). Само АЛУ и аккумулятор (АСС) — 32-разрядные. Процессор имеет аппаратную поддержку режима автоинкремента/автодекремента двух вспомогательных регистров (AR0, AR1), которые содержат адреса памяти данных при косвенном режиме адресации с целью оперирования массивами данных в одном командном цикле. Стек (STACK) состоит из четырех 12-разрядных регистров и может использоваться для сохранения 12 младших разрядов аккумулятора с помощью специальных команд либо для автоматического сохранения счетчика команд (PC) при прерывании или переходе к подпрограмме. Процессор имеет внешнее прерывание с жестко привязанным адресом перехода (02), внешний входной сигнал (BIO), который управляет переходом по специальной команде (BIOZ), — переход либо осуществляется, либо нет, в зависимости от уровня на входе BIO. Восемь 16-разрядных портов ввода/вывода служат для обмена данными с периферийными устройствами. Процессор может адресовать 4K 16-разрядных слов памяти программ, из них 1.5K слов может находиться в ПЗУ на кристалле процессора (режим микроЭВМ) либо все во внешней памяти (режим микропроцессора). Память данных состоит из 144 16-разрядных слов ОЗУ на кристалле процессора. При необходимости начальной инициализации данных массивы констант можно размещать в памяти программ и с помощью специальных команд обмена шины данных и шины команд переписывать их в память данных либо загружать из ПЗУ, адресуемое через порты ввода/вывода.

TMS320C14 содержит периферию, позволяющую использовать его как контроллер, управляющий событиями внутри какой-либо системы. Процессор содержит два таймера-счетчика, сторожевой таймер, один последовательный и семь 16-разрядных портов ввода/вывода. 'C14 имеет также менеджер событий, состоящий из сравнивающей и захватывающей частей, позволяющий производить действия без вмешательства процессора. Сравнивающая система со-





*процессор `C52 имеет один последовательный порт

Рис. 3. Структурная схема процессора TMS320C5X

приложениях (рис. 2). Специальный сопроцессор обеспечивает обработку 8- и 16-битовых видов передачи и прямую асинхронную связь с большинством микроконтроллеров. Устройство позволяет также производить аппаратное сжатие данных по μ-закону и A-закону, преобразование 8-битовой логарифмической ИКМ в линейную ИКМ. Процессор включает еще два независимых последовательных порта и таймер.

Процессоры второго поколения TMS320.2X имеют архитектуру, аналогичную первому поколению, но с более широкими возможностями. Первый процессор этого поколения — TMS320.20 не нашел своего потребителя и в настоящее время не выпускается, поэтому не будем на нем заострять свое внимание. Все процессоры второго поколения могут использовать 64K слов памяти программ и столько же памяти данных. Все имеют по шестнадцать 16-разрядных портов ввода/вывода, по одному последовательному порту и по одному таймеру. Количество внешних прерываний увеличено до трех, плюс прерывания от таймера и последовательного порта. У второго поколения процессоров появилась возможность реализации внешнего контроллера прямого доступа к памяти путем притормаживания работы процессора с помощью сигнала, подаваемого на специальный вход. Умножитель теперь может помимо операции умножения возводить в квадрат за один командный цикл. Появилась аппаратная поддержка повторения выполнения одной команды заданное количество раз. Количество вспомогательных регистров, используемых при косвенной адресации памяти данных, увеличено до восьми, появился режим двоично инверсной косвенной адресации, который очень полезен при реализации быстрого преобразования Фурье (БПФ). Теперь об особенностях каждого конкретного чипа.

TMS320C25 выпускается в модификации с командным циклом 80 и 100 нс. Процессор имеет 544 слова ОЗУ, состоящие из трех блоков, имеющих возможность различной конфигурации и 4K слов масочного ПЗУ.

TMS320E25 полностью повторяет процессор 'C25, но только имеет перепрограммируемое ПЗУ с ультрафиолетовым стиранием.

поставляет шесть регистров в каждом из таймеров. Если величины равны, то производится запрограммированное действие по сбросу или установке одного из внешних выводов сравнивающего устройства. Обычно это устройство используется для создания систем широтно-импульсной модуляции (ШИМ). Сравнивающая система имеет специальный режим, который повышает разрешение ШИМ до 40 нс. Система захвата взаимосвязывает действие с четырьмя различными событиями. Когда вывод захвата установлен, содержимое одного из таймеров записывается в четырехуровневый стек FIFO.

Процессор TMS320C15 позволяет наращивать возможности ядра 'C10. При производстве кристалла в нем могут быть прошиты схемы, разработанные заказчиком. Процессор также имеет 4K слов внутренней ПЗУ или внешней памяти программ и 256 слов ОЗУ данных на своем кристалле. Существуют как однократно программируемые, так и электрически перепрограммируемые версии кристалла. Процессор выпускается с 5- и 3,3-вольтовым (TMS320LC15) питанием.

TMS320C16 — самый быстрый из своего семейства процессор с командным циклом 114 нс. Имеет на кристалле 8K слов ПЗУ и 256 слов ОЗУ. Может использовать 64K слов внешней памяти.

TMS320C17 обладает всеми особенностями 'C15, но имеет специальную периферию, которая позволяет использовать процессор в телекоммуникационных

Тип процессора	Командный цикл (нс)	Память (слов)			Порты ввода/вывода			
		ОЗУ	ПЗУ	ППЗУ	Возможность адресации	Параллельный	Последовательный	Таймер
TMS320C10	200	144	1.5K		4K	8X16		
TMS320C10-25	160	144	1.5K		4K	8X16		
TMS320C10-14	280	144	1.5K		4K	8X16	1	4
TMS320C14	160	256	4K		4K	7X16	1	4
TMS320C14/P14	160	256		4K	4K	7X16		
TMS320C15/LC15	200	256	4K		4K	8X16		
TMS320C15-25	160	256	4K		4K	8X16		
TMS320E15/P15	200	256		4K	4K	8X16		
TMS320E15-25	160	256		4K	4K	8X16		
TMS320C17/LC17	200	256	4K		4K	6X16	2	1
TMS320E17/P17	200	256		4K	4K	6X16	2	1
TMS320C16	114	256	8K		64K	8X16		
TMS320P16	114	256		8K	64K	8X16		
TMS320C25	100	544	4K		128K	16X16	1	1
TMS320C25-50	80	544	4K		128K	16X16	1	1
TMS320E25	100	544		4K	128K	16X16	1	1
TMS320P25	100	544		4K	128K	16X16	1	1
TMS320C26	100	1.5K	256		128K	16X16	1	1
TMS320C28	80/100	544	8K		128K	16X16	1	1
TMS320C50	50/35/25	10K	2K		192K	64KX16	2	1
TMS320C50	50/35/25	1K	4K		192K	64KX16	1	1
TMS320C50	50/35/25	2K	8K		192K	64KX16	2	1
TMS320C50	50/35/25	4K	16K		192K	64KX16	2	1

TMS320P25 полностью повторяет процессор ‘C25, имея при этом однократно программируемое ПЗУ.

TMS320C26 включает в себя четыре блока ОЗУ (всего 1.5K слов), которые после конфигурации разделяются на память программ и данных.

TMS320C28 имеет 8K слов масочного ПЗУ и режим пониженного энергопотребления. В этом режиме потребляемый ток составляет 100 мА. Наиболее пригоден для применения в малогабаритных устройствах с автономным питанием.

Процессоры пятого поколения TMS320C5X также имеют архитектуру, в основном похожую на своих предшественников, но расширенную дополнительными возможностями (рис. 3). Производятся процессоры с 5- и 3-вольтовым питанием. Существует три режима энергопотребления: активный, периферийный и сонный. Активный режим потребления является основным; потребление в этом режиме составляет 1.5 мА/MIPS для 3В и 2.5 мА/MIPS для 5В. В периферийном режиме центральный процессор останавливается, работает только периферия, при этом потребление составляет 0.25 и 0.4 мА/MIPS для 3 и 5В. В сонном режиме все “засыпает” до возникновения внешнего прерывания, потребление снижается до 5 мкА. Процессор может использовать 244K слов памяти: 64K — память программ, 64K — память данных, 64K — порты ввода/вывода, 32K — глобальная память. При создании мультипроцессорных систем существует необходимость совместного использования участка памяти. Для этого в процессорах пятого поколения предусмотрены сигналы запроса и готовности при обращении к глобальной памяти, доступ к которой должен управляться специальным арбитром. Появилась аппаратная поддержка кольцевых буферов. В режиме косвенной адресации можно создать в памяти данных два независимых кольцевых буфера (удобно при реализации фильтра и коррелятора). При работе с ними начало и конец буферов в памяти отслеживаются автоматически с помощью специальных регистров. Существует аппаратная поддержка циклического повторения блока программы заданное количество раз. Появилось новое устройство, называемое параллельным логическим, позволяющее производить битовые и логические операции над операндами из памяти и различных регистров. В процессоре есть 11 теневых регистров, которые служат для автоматического сохранения состояния основных регистров процессора в случае возникновения программного или аппаратного прерывания. Для возможности работы с медленной памятью на кристалле процессора сделан программируемый генератор тактов ожидания. Количество тактов может быть установлено от 0 до 7. Из периферии процессоры пятого поколения имеют: 64K 16-разрядных портов ввода/вывода, один таймер, два последовательных порта, притом один из них может работать в режиме с временным разделением доступа (кроме ‘C52, у которого один последовательный порт). Процессоры имеют интерфейс JTAG. Различия между процессорами пятого поколения состоят лишь в конфигурации внутрикристальной памяти (см. таблицу). ОЗУ разделяется на несколько блоков, один из которых может принадлежать либо памяти программ, либо памяти данных, в зависимости от конфигурации. ПЗУ всегда принадлежит памяти.

Процессоры TI с фиксированной точкой нашли свое применение во многих отраслях электронной промышленности. На их основе созданы устройства различной сложности — от систем с одним “чипом” до многопроцессорных систем обработки аналоговой и цифровой информации. Сигнальные процессоры TI по праву занимают устойчивое положение на рынке электронных компонентов.

Прошлым летом ITU-TSS принят телекоммуникационный стандарт V.34, описывающий методы передачи данных со скоростью до 28 800 бит/с. Что делать покупателям, которым требуются высокоскоростные модемы? Воспользоваться возможностью модернизации до V.34, приобрести первый же модем с надписью «V.34», подождать появления более поздних и более совершенных моделей или ориентироваться на стандарты независимых поставщиков?

Стандарт V.34

Новое поколение модемов — новые проблемы выбора

Леонид Кузнецов

Принятый прошлым летом ITU-TSS (бывшей CCITT) стандарт V.34 описывает методы передачи данных со скоростью до 28,8 Кбит/с, что дает удвоение по отношению к скорости модемов предыдущего поколения (V.32bis), и таким образом позволит достичь верхней границы скоростей, теоретически достижимых на существующих телефонных линиях. С использованием встроенных алгоритмов компрессии модемы V.34 смогут достигать эффективной скорости передачи более 100 Кбит/с.

Уже ясно, что V.34 будет характеризоваться беспрецедентным соотношением "цена/производительность". Это обещает много преимуществ, но новые модемы, по крайней мере первое время, будут обладать и рядом недостатков.

В первую очередь встанет проблема взаимной совместимости ранних и поздних изделий, а также изделий разных производителей. При разработке предыдущего поколения модемов потребовалось от 6 до 12 месяцев для решения проблем взаимной совместимости с момента начала поставок изделий на рынок. В случае с V.34 может потребоваться еще больше времени.

Кроме того, последовательные порты IBM PC и Macintosh слишком медленны для V.34, и все еще нет согласия между производителями модемов относительно лучшей альтернативы.

Наконец, стандарт V.34 определяет только часть функциональных возможностей модемов как обязательные, оставляя другую часть факультативной. Поэтому разница между реализациями стандарта V.34 различными производителями может сохраняться достаточно долгое время.

▲V.34

Исследовательская группа SG 14 при ITU-TSS (ранее — SG XVII при CCITT) с 1991 года разрабатывала рекомендацию для дуплексного модема, работающего на скоростях выше скорости 14,4 Кбит/с, поддерживаемой стандартом V.32bis. В США работа курировалась комитетом TR-30.1 при Telecommunications Industries Association (TIA). Проект V.34 был закончен в феврале 1994 года, а в июне 1994 года состоялось его формальное утверждение в ITU-TSS.

В настоящее время рекомендация V.34 определяет скорости до 28,8 Кбит/с. Предполагалось, что окончательный вариант может включать скорости до 32 Кбит/с, что технически возможно. Независимые производители модемов, однако, сочли, что более высокие скорости лучше реализовать в фирменных стандартах, надеясь получить преимущество в конкуренции.

▲От Bell-103 до V.34

Пропускная способность старых модемов ограничивалась их неспособностью подстраиваться под характеристики телефонных линий. Даже если на определенных частотах имелись помехи, модемы вынуждены были использовать одну фиксированную несущую частоту. Модемы V.34 смогут выбирать оптимальную несущую частоту и достигать более высокой пропускной способности за счет точной подстройки к характеристикам телефонного канала.

Собственно, вся история модемной техники представляет собой историю усовершенствований и нововведений. В погоне за пределом совершенства менялись конструкции модемов и вводились новые методы. Но совершенство, конечно, недостижимо, и понятие лучшего изделия верно только до очередного технологического прорыва.

Вначале был Bell 103 (300 бит/с), с частотной модуляцией, использовавшей разные частоты для передачи логических «1» и «0». Bell-103 поддерживал дуплексный протокол, причем для передачи данных в разных направлениях использовались различные пары частот.

Bell 212A (1200 бит/с) использовал фазовую модуляцию. Этот метод применяет для передачи нулей и единиц сдвиг по фазе одной и той же несущей волны.

V.22bis ввел 16-позиционную квадратурно-амплитудную модуляцию и адаптивную коррекцию АЧХ (модем анализирует принимаемый сигнал и подстраивает фильтр при-

емника для обеспечения наилучшей работы) для достижения скорости 2400 бит/с в полном дуплексе. Каждый символ представляется как величина амплитуды и относительный сдвиг фазы принимаемого сигнала в определенный момент времени.

V.32 (9600 бит/с) и V.32bis (14 400 бит/с) добавили метод эхокомпенсации, благодаря которому при передаче данных в каждом направлении могла использоваться полная полоса канала. Решетчатое кодирование сделало практичным использование 32-точечной модуляции.

Наконец, V.34 (28 800 бит/с) вводит ряд новых методов коррекции, кодирования, управления уровнем сигнала и кадрирования данных. Перечислим эти нововведения.

1. Оптимальная — круговая граница сигнального пространства.
2. Кадрирование данных распределяет биты по нескольким символам. Это улучшает поддержку разнообразных комбинаций модуляционных скоростей и скоростей передачи данных, а также делает возможным использование вторичного канала.
3. Многомерное представление модуляционного пространства и кодирование данных с использованием разделения модуляционного пространства на кольца с равным числом точек.
4. 4-мерное решетчатое кодирование (64-, 32- и 16-позиционный конволюционный код) повышает помехоустойчивость за счет более эффективного использования сигнального пространства.
5. Новая форма линейной адаптивной коррекции — предкоррекция, которая оказывает оптимизирующее влияние не только на принимаемый, но и на передаваемый сигнал путем компенсации неравномерности АЧХ канала. Для измерения АЧХ телефонного канала между модемами передается тестирующий сигнал в диапазоне 150-3750 Гц. После получения данных о характеристиках канала приемники и передатчики в каждом из модемов начинают работать вначале на минимальной скорости, затем наращивают ее до максимально возможной при данной обстановке.
6. Прекодирование — нелинейная прекомпенсация для уменьшения шума эквалайзера, вызванного нелинейными искажениями.
7. Нелинейное кодирование деформирует расположение внешних точек для достижения неоднородного распределения вероятности ошибки, что повышает устойчивость к нелинейным искажениям.
8. Адаптивное регулирование уровня передаваемого сигнала дает наилучшую производительность в условиях искаженной передачи в канале. Этот метод также можно использовать в сотовой связи.
9. Масштабирование поддерживает наилучший уровень передаваемого сигнала при использовании разных методов модуляции.
10. Выбор оптимальной несущей частоты в диапазоне 1600-1959 Гц.

▲V.34 и V.8

При установлении связи одного модема с другим прежде всего происходит процедура установления соединения, так называемое «рукопожатие» для согласования используемых схем модуляции. Старые модемы используют процедуры установления соединения, базирующиеся на тональных характеристиках каждой из последовательностей согласования протоколов. Тональное распознавание требует много времени, чем больше поддерживается методов модуляции, тем больше времени требуется на распознавание.

Нынешние высокоскоростные модемы поддерживают до шести методов модуляции, и для правильного распознавания несущей частоты иногда требуется до 8 секунд. Общая же продолжительность процедуры установления соединения, включая установку режимов коррекции ошибок и сжатия, занимает 20-30 секунд.

Модемы V.34 поддерживают гораздо больше методов модуляции. Они должны быть совместимы не только с большим семейством дуплексных протоколов для модемов и с факс-аппаратами группы 3, но и с новыми высокоскоростными факсимильными приложениями, оборудованием сжатия речи, работающим на ИКМ-каналах (64 Кбит/с), а также поддерживать видеотелефон и другие новые виды связи. Поэтому V.34 требует иного подхода к процедуре установления соединения. Этот стандарт частично использует рекомендацию, находящуюся сейчас в стадии разработки: V.8 (ранее назывался V.id).

Перед тем как начинается установление соединения в соответствии с V.34, происходит установление соединения по V.8. В первую очередь модемы определяют список функциональных возможностей, которые являются общими для обоих модемов. После этого тестирующим сигналом определяются характеристики телефонного канала, и тогда приемники и передатчики в каждом из модемов начинают работать вначале на минимальной скорости, затем наращивают ее до максимально возможной. Вся процедура укладывается в пять секунд.

Хотя никто из производителей не огласил планов использования V.8 для не-V.34-модемов, ничто не препятствует использованию V.8 совместно с модемами V.32bis (или с какими-либо еще), для того чтобы уменьшить время установления соединения.

▲V.34 и V.42bis

Спецификация V.34 в основном описывает процессы модуляции.

Компрессия данных описывается отдельным стандартом протокола V.42bis, работающим совместно с протоколом коррекции ошибок V.42. Комбинация V.42—V.42bis является весьма эффективной, и поэтому она наверняка будет реализована большинством производителей во всех модемах V.34.

V.42bis дает при благоприятных условиях коэффициент компрессии до 4. При физической скорости передачи данных 28,8 Кбит/с скорость некомпрессированных данных в интерфейсе DTE составит 115,2 Кбит/с, что подразумевает очень высокие требования к быстродействию алгоритма компрессии. При полном дуплексе потребуется такое же быстродействие для декомпрессии данных.

Производителям модемов придется пойти либо на ограничение пропускной способности модема, либо на применение более быстрых процессоров и увеличение их количества в конструкции модема. Вероятно, оптимальным здесь является применение специализированных БИС, однако производители должны в любом случае оценить оправданность затрат.

▲V.34 и EIA-232

Компьютеры IBM PC и Macintosh используют последовательный интерфейс в комбинации с UART (универсальный асинхронный приемопередатчик) и символьно-ориентированным программным обеспечением, которое перестает надежно работать на скоростях выше 20 Кбит/с, что значительно меньше, чем позволяет V.34 (особенно в комбинации с V.42bis). Даже если в компьютере установлен UART 16550, который обеспечивает буферизацию данных, коммуникационное программное обеспечение может оказаться неприспособленным к использованию буфера и постоянному обслуживанию UART в многозадачной среде.

Следовательно, для пользования модемами V.34 потребуются новые интерфейсы. Существует по крайней мере пять возможных решений.

Первое из них — это интерфейс EIA-530A (2,1 Мбит/с), разработанный несколько лет назад для того, чтобы заменить как EIA-232, так и V.35. EIA-530A совместим сверху вниз с EIA-232, так что существующие модемы и кабели могут с ним работать. К несчастью, EIA-530A не получил заслуженного признания в промышленности IBM PC-совместимых компьютеров. Если EIA-530A и найдет применение с модемами V.34, то только с терминальными процессорами, концентраторами и маршрутизаторами.

Вторую возможность дает V.10 (100 Кбит/с). Этот стандарт электрически совместим с EIA-232. Проблема заключается в том, что пока V.10 будет реализовываться в последовательных интерфейсах, порты UART и программное обеспечение останутся неизменными, так что работа на скоростях выше 20 Кбит/с все еще будет проблематичной.

Третий подход предлагает использовать в модемах V.34 стандартный двунаправленный параллельный порт IEEE 1284. Для подключения такого модема к компьютеру можно будет использовать стандартный двунаправленный параллельный порт, который есть в каждом PC. Такое подключение обеспечивает передачу информации на скоростях до нескольких мегабит в секунду без потери данных даже при работе в многозадачных операционных системах. Фирма Xircom Inc. объявила о выпуске программного обеспечения под Windows и DOS для перенаправления данных с последовательного порта на параллельный. Компания Microcom Inc. представила аналогичное программное обеспечение для Windows. Эти программные продукты, однако, не являются универсальными решениями.

Четвертая возможность связана с интерфейсом PCMCIA, который сегодня имеется почти на каждом переносном компьютере (но не на настольных PC). Можно ожидать этого решения, когда удастся реализовать V.34 в габаритах PCMCIA.

Наконец, пятое решение связано с реализацией фирменных улучшенных последовательных интерфейсов, снабженных дополнительной картой с буферами и программными драйверами для прямого подключения к шине AT.

Большинство выпускаемых в настоящий момент модемов V.fast и V.34 имеют только последовательный порт со скоростью 115,2 Кбит/с. Однако некоторые производители планируют выпуск модемов с последовательными портами на скорости до 230,4 Кбит/с и специальной функцией усреднения скорости потока данных, позволяющей использовать пропускную способность последовательного порта, соответствующую усредненной скорости потока данных в канале без риска потери данных. Microcom и ZyXEL Communications предполагают снабжать свои модемы также и параллельным портом.

▲V.34 и V.fast

С модемами V.34 легко спутать три типа фирменных высокоскоростных модемов. Во-первых, это V.fast. V.fast — общепринятое название стандарта дуплексных модемов для 2-проводных коммутируемых и выделенных линий со скоростью более 19,2 Кбит/с. Предполагается, что в ряде случаев они могут быть модернизированы до V.34. Во-вторых, многие модемы с фирменными протоколами на 19,2 Кбит/с и более также называются модемами «V.fast», хотя не имеют ничего общего с V.34. В третьих, существуют также расширения V.32bis в виде V.32terbo, которые обеспечивают скорости выше чем 14,4 Кбит/с.

В конце 1993 года отделение Rockwell International Corp. представило свою реализацию набора микросхем класса V.fast — V.FC. Набор является фирменной реализацией версии V.fast, работающей на скоростях до 28 Кбит/с. Эта реализация, поддерживаемая рядом производителей, включая Hayes и Microcom, однако, не использует процесс установления соединения по стандарту V.8, так что без модернизации она не может быть полностью совместима с будущими модемами V.34.

Весной 1994 года многие фирмы, например Practical Peripherals, Supra, Zoom, не только продавали модемы класса V.FC, выполненные также на базе набора RC288ACi фирмы Rockwell, но и рекламировали, что они будут доведены до стандарта V.34. Для покупателей этих модемов оказалось неприятным сюрпризом, что обещанная модернизация модемов, во-первых, будет стоить от 50 до 70 долларов, во-вторых, клиент за свой счет должен отослать модем к производителю. При этом сроки выполнения модернизации остаются неизвестными. Есть все основания предполагать, что процесс модернизации из V.FC до V.34 на деле окажется значительно сложнее, чем ожидалось. Могут потребоваться достаточно существенные аппаратные модификации.

Свои фирменные модемы V.fast для коммутируемых линий представили также Motorola, US Robotics, AT&T, Telebit, Racal-Datacomm, General Datacomm. Многие из названных разработчиков заявили, что их изделия будут модернизируемы до V.34. Пока лишь Courier V.Everything фирмы US Robotics оказалось возможным модернизировать до V.34 путем замены микропрограммы. Впрочем, аппаратная база этих модемов была полностью завершена до окончательного принятия стандарта V.34, поэтому неизвестно, не потребуется ли в дальнейшем замена каких-либо аппаратных средств.

▲V.34 и... V.34

Так как стандарт V.34 описывает часть функций модемов как дополнительные, очевидно, что модемы V.34 различных производителей могут значительно отличаться в части, относящейся собственно к процессу модуляции-демодуляции. Вопрос в том, насколько существенны эти отличия.

В отличие от всех предыдущих стандартов, опирающихся на модуляционные скорости не более 2400 бод, V.34 реализует кроме 2400 и другие обязательные к применению модуляционные скорости: 3000, 3200 бод и дополнительные: 2743, 2800, 3429 бод. Из списка рекомендованных модуляционных скоростей модем V.34

может поддерживать не все модуляционные скорости. Чем грозит отсутствие в модеме некоторых модуляционных скоростей?

Такой модем не при всех условиях будет способен поддерживать все скорости передачи данных. В соответствии с V.34 скорости передачи данных в основном канале лежат в диапазоне 2,4-28,8 Кбит/с с шагом 2,4 Кбит/с, при этом не при всех модуляционных скоростях могут быть реализованы любые скорости передачи данных. Например, максимальная скорость передачи данных при модуляционной скорости 2400 бод составляет 21,6 Кбит/с, а скорость 24 Кбит/с может быть достигнута при модуляционной скорости не менее 2743 бод. Следующая обязательная к применению модуляционная скорость равна 3000 бод, что требует приблизительно на 250 Гц более широкой полосы пропускания частот канала передачи данных. Поэтому может оказаться, что модем с неполным набором модуляционных скоростей будет способен работать при ограниченной полосе пропускания канала только на более низкой скорости 21,6 Кбит/с, в то время как этот канал мог бы обеспечить скорость 24 Кбит/с.

Модемы V.34 также могут поддерживать не все режимы и дополнительные возможности, в том числе 64- и 32-позиционный конволюционный код, вторичный канал 200 бит/с, асимметрию скорости передачи данных в разных направлениях, не все несущие частоты.

Применение 64- и 32-позиционного конволюционного кодирования повышает помехозащищенность, однако требует дополнительной вычислительной мощности. Можно с уверенностью сказать, что не все производители модемов решатся реализовать аппаратную часть модемов с применением самых мощных на сегодняшний день сигнальных процессоров, а те, кто на это пойдет, рискуют проиграть за счет более высокой цены. Кроме того, процесс устранения ошибок в микропрограммах может значительно затянуть полную реализацию всех возможностей и свести таким образом на нет имеющиеся преимущества.

Асимметрия скорости передачи данных может стать значительным преимуществом при передаче данных по асимметричному каналу. Асимметрия канала — достаточно распространенное явление, особенно в телефонных сетях с устаревшим или плохо обслуживаемым оборудованием. Неспособность модема работать в асимметричном режиме может являться, таким образом, недостатком, не позволяющим полностью приспособиться к конкретной линии. В этом случае скорость передачи данных в обоих направлениях будет определяться худшими условиями передачи.

Вторичный канал 200 бод, специфицированный в V.34, открывает естественный путь к построению систем управления распределенными сетями передачи данных. Современные (даже не V.34) модемы ведущих производителей в той или иной степени реализуют вторичные каналы передачи данных. В основном эти каналы используются для передачи информации о состоянии модемной сети и управления ею. Как правило, в таких сетях используются профессиональные модемы, поэтому следует ожидать, что появятся модемы V.34 и в исполнении со вторичным каналом.

Напомним: для того чтобы наилучшим образом использовать пропускную способность телефонных каналов, модемы V.34 могут, в частности, выбирать оптимальную несущую частоту в диапазоне 1600-1959 Гц. Не все модемы будут предусматривать работу на любой из специфицированных несущих частот, что приведет к неполному использованию возможностей отстройки от помех и неоптимальному размещению спектра сигнала в полосе пропускания канала.

V.34: настоящее или будущее?

Все изложенное позволяет сказать с уверенностью только одно: чем совершеннее предлагаемые решения, тем больше неопределенности в том, насколько те или иные конкретные изделия реализуют их.

Сейчас рождается новое поколение модемов. Технологический разрыв по отношению к предыдущему поколению весьма велик, реальные же преимущества менее значительны. При переходе от стандарта V.22bis к V.32bis скорость передачи данных увеличилась в шесть раз, а при переходе от V.32bis к V.34 — только в два. Технология V.34 вплотную подошла к физическому пределу скорости передачи данных по телефонным каналам и является очень зависимой от качества телефонного соединения, что может не позволить использовать все ее преимущества в полном объеме. Дальнейшее существенное увеличение скорости передачи данных будет возможно только за счет использования цифровой технологии телефонных коммуникаций, в частности, за счет развития сетей ISDN.

Остается еще один вопрос, быть может самый главный: сколько времени потребуется на устранение неизбежных ошибок в микропрограммах. В условиях спешки ошибки могут сделать ранние выпуски модемов V.34 если не бесполезными, то по крайней мере значительно менее надежными, чем существующие высокоскоростные модемы.

При всем этом нет никакого сомнения в том, что модемы V.34 имеют большое будущее. Так или иначе, последнее слово остается за потребителем, и ему виднее, спешить за V.34 или нет.

Рабочая станция в «коробке от пиццы»

Наталия Ефремова, Олег Татарников

Основанная в 1982 году фирма Silicon Graphics Inc. (SGI) уже с конца 80-х годов является безусловным лидером на рынке профессиональных рабочих станций для компьютерной графики. Подавляющая доля профессиональной компьютерной трехмерной анимации делается сегодня на этих станциях. Все крупнейшие кино- и телестудии мира, занимающиеся трехмерной анимацией, имеют компьютеры SGI. Большинство ведущих производителей программного обеспечения для трехмерной графики используют эту платформу если и не как единственную, то как основную. Библиотека IRIS Graphics Library фактически стала стандартом построения high-end графических приложений.

Однако стремительное развитие персональных компьютеров, появление высокоскоростных графических акселераторов, недорогих дисководов CD-ROM, высококачественных аудиоплат и другого оборудования, а главное — изощренных программ для графики и анимации на других платформах заставили Silicon Graphics искать новые пути для сохранения своих позиций. И в результате, наряду с созданием мощнейших многопроцессорных POWER ONYX (по некоторым параметрам сравнимых с суперкомпьютерами CRAY) и расчетных суперсерверов CHALLENGE, с дальнейшим развитием настольных станций INDIGO2, SGI разработала и выпустила в 1993 году на рынок малютку INDY, обладающую вычислительной мощностью профессиональной рабочей станции и габаритами коробки от пиццы (ее корпус так и называется — Pizza-Box). Но самым важным ее параметром является цена, сравнимая с ценой сопоставимых по конфигурации PC. Подобное соотношение "цена/производительность" делает "малютку" весьма привлекательной не только в таких традиционных для SGI областях, как графика, анимация и спецэффекты для видео и кино, но и во многих приложениях, характерных ранее для Mac и PC: настольных издательских системах, сетевых и коммуникационных задачах, обучающих системах и др. Появление этой станции дает еще больше оснований утверждать, что путь профессионала — все-таки Silicon. И совсем не потому, что модное выражение "силиконовая графика" стало автоматическим определением некоего high-класса компьютерной анимации, как скажем Mersedes или BMW для "новых русских" в автомобилестроении.

Итак, что может "коробка из-под пиццы".

▲Центральный процессор

Самым "медленным" процессором для INDY на сегодняшний день является R4600PC — 64-разрядный 100 МГц RISC-процессор архитектуры MIPS.

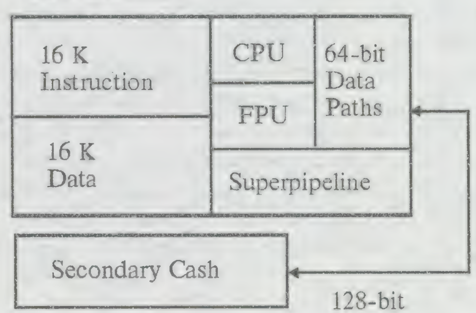
Далее — по таблице:

INDY CPU Performance Chart			
	R4600PC 100 МГц	R4600SC 133 МГц	R4400SC 150 МГц
Cash(primary)	16K data 16K instruction	16K data 16K instruction	16K data 16K instruction
Cash(secondary)	—	0.5Мбайт	1Мбайт
SPECint92	62.8	93.7	91.7
SPECfp92	49.9	72.9	97.5
AIM	63.7	104	113.3
DhryMIPS	119	163.5	136
MFLOPS(100x100 DP LINPACK)	11	19	25

Напомним, что Pentium имеет два 8K кэша, а 486DX — 8K на все. Хотя Pentium и приближается по производительности к этим процессорам (его производительность превышает 100 MIPS), поддержка кодов всех предыдущих процессоров фирмы Intel не может не внушать опасения за его архитектуру. Как лабиринт Минотавра, он, похоже, имеет закоулки, куда не забредает ни одна "шальная" команда.

Процессорный модуль можно легко заменить на более мощный по мере возрастания вычислительных запросов, а также с появлением новых технологий MIPS. Например, уже появился процессор R4400 200 МГц, который доступен пока только для старших моделей SGI, есть новая разработка MIPS — процессор R8000(TFP), по плавающей арифметике в 5-6 раз превосходящий R4400 (для POWER-моделей SGI).

Внутренняя структура процессорного модуля:



▲ОЗУ

В минимальной конфигурации INDY имеет 16 Мбайт RAM. Можно использовать стандартные 72-pin SIMMS-модули, устойчиво работающие на частоте 50 МГц. 32-мегабайтными модулями можно нарастить память до 256 Мбайт.

▲Шина

В отличие от архитектуры PC, где системная шина общая для связи и с оперативной памятью, и со слотами расширения, на INDY имеется отдельно подсистема памяти с обменом до 400 Мбайт/с (50 МГц) и отдельно 64-разрядная системная шина расширения с обменом до 267 Мбайт/с (33 МГц). Напомним, что декларированная пропускная способность лучших 64-разрядных шин PCI (Peripheral Component Interconnect) и VESA 2 (Video Electronic Standards Association) для PC — до 132 Мбайт/с, однако PC-контроллеры разрабатываются преимущественно под 32-разрядные шины PCI (до 30 Мбайт/с) и VL-bus (до 20 Мбайт/с) соответственно. Стандартная до последнего времени 16-разрядная шина ISA (Industry Standard Architecture) обеспечивала лишь 5 Мбайт/с, а большинство существующих PC-контроллеров рассчитаны именно на этот стандарт.

▲Жесткий диск

Базовые внутренние жесткие диски, поставляемые с INDY, рассчитаны на встроенный контроллер Fast SCSI-2 и обеспечивают время доступа до 9 мс и скорость непрерывной передачи данных до 5 Мбайт/с. Можно установить внутрь до 2 Гбайт, а более емкие и скоростные диски — внешними (всего до 7 SCSI-устройств на одном контроллере). Причем можно использовать любые SCSI-диски.

▲Видеоподсистема

Видеосистемы SGI до сих пор считаются непревзойденными, а базовая для INDY графическая система XL является оптимальной для плоской графики. Для 3D-графики предпочтительнее более дорогая система — XZ с аппаратным Z-буфером, который используется для удаления невидимых линий, закраски поверхностей и других аппаратных графических примитивов, ускоряющих отрисовку экрана. Остановимся подробнее на базовой XL-системе (вся 3D-графика SGI на ней полностью эмулируется программно). Существуют две модификации: 8 бит на цвет (виртуально 24 с оптимизацией палитры) и 24 бита на цвет (16,7 млн. цветов, одновременно отображаемых на экране). Ни один из существующих и проектируемых графических акселераторов для PC не может сравниться ни по скорости, ни по возможностям с графическими системами SGI.

INDY XL Graphics Performance			
	R4600PC @100 MHz	R4600SC @133 MHz	R4400SC @150 MHz
XLines/sec	1.4M	1.5M	1.6M
3D Vectors/sec	333K	443K	800K
Tmesh/sec(z)	33K	51K	64K
Screen Clear	3ms	3ms	3ms
DMA Pixel Xfer			
8-bit	45M	N/A	51M
24-bit	19M	N/A	40M

Кроме того, INDY имеет 3 видеовхода: композитный (PAL/NTSC низкочастотный VHS) вход, S-VHS-вход, цифровой (YUV) SGI-видеовход, а также гнездо для стереоочков. В комплекте с INDY поставляется цветная цифровая видеокамера (с разрешением 640x480), под управлением пакета InPerson можно проводить видеоконференции с удаленными станциями. Дополнительно приобретается видеоплата IndyVideo для ввода/вывода видеоизображения, изготовления обучающих кассет, презентаций и демонстраций в режиме реального времени. Добавление к этой плате модуля Cosmo Compress (motion JPEG видеокомпрессия/декомпрессия) дает возможность оцифровки/записи полноформатного видео в режиме реального времени, нелинейного монтажа и спецэффектов.

Работа с видеовходами/выходами, а также с внешними видеоустройствами поддерживается базовыми средствами операционной системы.

▲Монитор

В самой дешевой комплектации с INDY поставляется 15" 70 Гц цветной монитор с разрешением 1024x768 семейства Sony Trinitron. Модели Trinitron отличаются высокой насыщенностью цвета и большой резкостью изображения. В стандартную комплектацию для INDY входит 17" цветной мультичастотный стереомонитор с разрешением 1280x1024. Можно заказать еще 20" стереомонитор или заменить (upgrade) 17" на 20" (1750 долларов) в дальнейшем. Следует подчеркнуть, что стереоочки работают только с 17" и 20" мониторами. Можно приобрести также и специальную цветную жидкокристаллическую панель (12" активная матрица LCD), которую можно использовать как плоский экран, а при снятой задней панели — как слайд для проекционного устройства.

▲Аудиоподсистема

Встроенная аудиосистема INDY обеспечивает DAT-качество звука с частотой дискретизации 48 кГц и имеет пять аудиопортов: цифровой стереовход/выход (AES/EBU), аналоговый стереовход, аналоговый стереовыход, вход для моно/стереомикрофона (douномикрофон входит в комплект поставки) и выход для стереонаушников. Два 16-разрядных стереоаудиокодера (дельта-сигма) обеспечивают 64х-оверсамплинг. Работа с аудиовходами/выходами, а также с внешними аудиоустройствами поддерживается базовыми средствами операционной системы. Можно записать звук в файл, отредактировать и затем переписать на носитель. Заявления же о том, что некоторые звуковые платы PC могут обеспечить запись на жесткий диск с CD-качеством, проверки не выдерживают[1].

▲Периферия

Дисковод CD-ROM для серьезной работы давно не роскошь, особенно в сфере графических приложений. На CD-ROM поставляются многочисленные собрания готовых рисунков, текстур, шрифтов и фотографий, а также дистрибутивы программ (для многих приложений это уже единственный носитель). Кроме того, сама операционная система IRIX для компьютеров SGI поставляется на CD с возможностью непосредственной загрузки из mini-root. Объем хранения на CD-дисках до 680 Мбайт. Кроме того, в операционной системе предусмотрена поддержка аудио-CD с возможностью оцифровки музыки в файл с последующим редактированием и перезаписью на цифровой или аналоговый носитель. Без проблем читаются CD-диски и Mac-, и PC-, и PhotoCD-форматов.

Встроенный интерфейс Fast SCSI-2 позволяет подключить CD-дисковод к цепочке из нескольких устройств. Из моделей, наиболее полно поддерживающих эти функции на INDY, можно рекомендовать две:

1. Toshiba XM3401B (время доступа — 200 мс, передача данных — 330 Кбайт/с).
2. Sony CDU-561 (время доступа — 295 мс, передача данных — 300 Кбайт/с).

Очень полезное устройство для INDY — DAT (Digital Audio Tape), которое можно использовать одновременно и как стример с объемом хранения до 2 Гбайт и скоростью передачи данных 183 Кбайт/с, и как цифровой аудиомагнитофон с возможностью оцифровки звука в файл с последующим редактированием и перезаписью на цифровой или аналоговый носитель. Причем данные можно хранить на обычных 4-миллиметровых аудио-DAT-кассетах. Некоторые производители программного обеспечения используют кассеты DAT или DDS (Digital Data Storage) в качестве носителей для своих дистрибутивов.

Можно использовать DAT-устройства и большей емкости (до 16 Гбайт), однако для них необходимы специальные кассеты DDS 2 и функции цифрового аудиомагнитофона теряются. Поэтому предпочтительнее 2-гигабайтные устройства, например модель Python фирмы Conner.

Полезным устройством для INDY является внутренний 21-мегабайтный магнитооптический дисковод (Floptical Disk Drive), если учесть, что на нем можно использовать и обычные 3,5" дискеты, а операционная система IRIX поддерживает файловые структуры MAC и PC.

Каждый компьютер INDY полностью укомплектован для включения в локальную сеть Ethernet (LAN) с TCP/IP-протоколом и глобальную сеть ISDN (WAN). Имеется встроенный адаптер Ethernet с портом 10 Base-T и гнездом AUI, встроенная поддержка стандарта ISDN для подключения к цифровым коммуникациям в глобальной сети, поддержка промышленного стандарта для сетевого программного обеспечения.

Сетевое программное обеспечение включено в операционную систему. INDY также имеет два последовательных порта RS-422 со скоростью обмена до 38,4 Кбод, двунаправленный параллельный порт high density Centronics, синхронно/асинхронный вход Fast (10 Мбайт/с) SCSI-2 и два слота расширения GIO-32-bis. Мышь и клавиатура стандартные, как на PC, с разъемом PS/2.

Поддерживаются также дигитайзеры (например, модели фирм Toshiba и Kurta) и чувствительные к нажатию планшеты (фирмы Wacom). Специальные устройства ввода — Dials, Dials and button, характерные для компьютеров SGI, облегчают работу при моделировании сложных объектов.

▲Операционная система

Базовой операционной системой на PC до сих пор остается 16-разрядная MS-DOS, зачастую являясь самым "узким" местом при увеличении производительности компьютера. Хотя Windows for Workgroups 3.11 в значительной мере (за исключением ядра) состоит из 32-разрядных команд, она все же продолжает оставаться оболочкой над MS-DOS, а под управлением 32-разрядной Windows NT программы для Windows вообще выполняются на 20% медленнее, чем в Windows 3.1.

[1] Мне кажется, системы Digidesign для Mac и PC выдержат такую проверку /Ред./

Компьютер INDY на сегодняшний день имеет 32-разрядную операционную систему IRIX 5.3 (это UNIX System V.4 с расширениями 4.3 BSD и SGI и оконной системой X-Window X11R5), а на старших моделях SGI с новым процессором R8000 (TFP) применяется 64-разрядная OS IRIX 6.0. Кроме того, сравнение UNIX со связкой DOS—Windows по ряду причин вообще неуместно. Реальная многозадачность, позволяющая во время работы переходить из одного приложения в другое, открывать и закрывать их, безболезненно "убивать" зависшие, не боясь потери данных, не может оставить равнодушным пользователей Windows, где предложение прервать зависшее приложение реально означает сброс всей системы. Система IRIX надежна и устойчива в работе и позволяет успешно справляться с аварийными ситуациями.



В качестве интерфейса с пользователем применяется очень удобная оболочка Indigo Magic, реализующая так называемый Media User Interface, когда пользователь работает с данными, которые в зависимости от типа отображаются соответствующей пиктограммой (например, уменьшенная картинка для изображения), а соответствующая прикладная программа вызывается автоматически в отдельном окне. Скажем, для работы с текстом достаточно выбрать его символическое изображение на экране, и текстовый редактор загрузится. Поддерживается работа с видео- и аудиоустройствами, видеокамерой и другой периферией. Базовые средства системы позволяют редактировать звук и видеокадры/последовательности. Интуитивный, дружественный интерфейс, ориентированный на ввод с помощью мыши (Drag&Drop), подобная макинтошевской "мусорная корзина", простая система управления позволяют пользователю

VENDOR	SGI	SGI	SGI	SGI	HP	COMPAQ	TANGENT	ALR	DEC
PRODUCT	INDY	INDY	INDY	INDY	Vectra	ProLin.	PCI 5100	Ev.VQ66	pcXL566
CPU	R4600PC	R4600PC	R4600SC	R4400	DX4	DX4	PENTIUM	PENTIUM	PENTIUM
FREC.	100 МГц	100 МГц	133 МГц	150 МГц	100 МГц	100 МГц	100 МГц	66 МГц	66 МГц
CASH	-	-	512 Кбайт	1 Мбайт	256 Кбайт	256 Кбайт	512 Кбайт	512 Кбайт	256 Кбайт
VIDEO	XL 8-bit Virtual 24	XL 24-bit	XL 24-bit	XL 24-bit	Cirrus Logic	Compaq QVision	Weitek	ATI	Weitek
RAM	16 Мбайт	32 Мбайт	32 Мбайт	32 Мбайт	16 Мбайт	16 Мбайт	32 Мбайт	16 Мбайт	16 Мбайт
HARD	-	1 Гбайт	1 Гбайт	1 Гбайт	540 Мбайт	525 Мбайт	1 Гбайт	-	1 Гбайт
SPECint	62.8	62.8	93.7	91.7	45.68	37.68	75.65	61.49	51.57
SPECfp	49.9	49.9	72.9	97.5	23.13	21.09	62.59	52.07	47.56
MONITOR	15"	17"	17"	17"	15"	15"	17"	15"	15"
CD-ROM	-	-	Toshiba 3401B	Toshiba 3401B	-	-	Plextor DM3028	-	Toshiba 4101B
PRICE	$5,744	$13,219	$19,681	$23,706	$4,427	$4,120	$6,736[1]	$8,310	$6,106

[1]Цены в США (к сожалению этот лучший в своем классе компьютер в России никто не продает). Все остальные цены — дилеров в России (CIF, Москва).

работать, практически не используя текстовых команд. Весь интерфейс полностью настраивается. Всем объектам можно задать свои пиктограммы и программы реагирования на выбор. Пользователь может набрать любое окружение в зависимости от задачи, сохранить его и загружать по мере необходимости. Пусть пользователи PC не боятся: там все, как и в Windows, только гораздо удобнее.

Доступ к системным функциям реализует System Manager, интерактивная оболочка, облегчающая конфигурирование и настройку. Гипертекстовый On-line Help и подробная документация в машиночитаемом виде (на системном CD-ROM) облегчают освоение операционной системы и среды окружения. Имеется расширенная программа диагностики.

IRIS Showcase — специальное программное обеспечение для изготовления демонстраций, презентаций и обучающих курсов, объединяющее 3D-модели, 2D-графику, текст и аудио/видеоинформацию. IRIS Explorer — программа для изготовления качественных 2D/3D-картинок, заставок и иллюстраций. Media Mail — поддержка функций электронной почты с передачей текста, картинок и аудио/видеоинформации.

INDY поддерживает файловые системы PC и Macintosh, имеет возможность работать с Novell NetWare и AppleTalk и выполнять все PC- и MAC-приложения в режиме эмуляции. Стоимость операционной системы и оболочки, кстати, входит в общую стоимость компьютера.

Сравнительная таблица по соотношению "цена/производительность" SGI INDY и лучших на сегодняшний день моделей PC (информация по лучшим персональным компьютерам взята из журнала BYTE за август 1994 года) приведена выше.

В комплектацию компьютеров, поставляемых без мониторов, включены мониторы SONY Trinitron. К ценам PC необходимо прибавить стоимость видео-, аудио- и сетевого оборудования.

Справедливости ради отметим, что сама по себе покупка станции SGI не означает, что ваша продукция немедленно и автоматически оставит позади сделанное конкурентами. Мы могли бы привести массу примеров, когда анимацию, сделанную на PC, не только трудно отличить от "силиконовой", но даже лучшие ее образцы превосходят по уровню многие работы на станциях SGI. Все, в конечном счете, определяется уровнем профессионализма аниматоров. Заборы можно красить и колонковыми кистями, а приобретение новых возможностей само по себе не рождает шедевров.

В заключение мы хотим остановиться на некоторых нетрадиционных до последнего времени применениях станций SGI, например в области издательских систем. Приведем решение фирмы COLORBUS — Cyclone PS PrintStation.

Рабочая станция с RISC-скоростным Adobe Photoshop (SGI-версия), сканер, принтер, сервер и RIP-конвертер одновременно в вышеупомянутой "коробке из-под пиццы". SGI INDY, сканер, принтер и фотонаборная машина обслуживают сеть рабочих мест PC и MAC. Полный контроль за прохождением запросов, приход/ожидание, распределение приоритетов, "замораживание" и снятие задач осуществляются интерактивно. Информационное окно каждой задачи позволяет детально отслеживать все этапы прохождения процесса. Постоянно "живущее" окно RIP-процесса облегчает поиск и отладку ошибок исходного PostScript-файла. Тем самым резко снижается количество непроизводительных затрат (и времени, и ресурсов) и уменьшается срок выдачи готовой продукции. Настройка и коррекция сканирования производятся либо непосредственно, либо из Adobe Photoshop при помощи специальных модулей plug-in. Программное обеспечение работает со специальным измерителем плотности цвета X-Rite DTP32, включает PostScript Level 2 Color Rendering Dictionaries и производит ручную или автоматическую цветокоррекцию. Программы скоростной печати Cyclone PS PrintStation, работающие как непосредственно, так и из Photoshop, позволяют обходить это слабое место всех настольных издательских систем.

Возможны следующие аппаратные конфигурации такой системы:

1) наиболее дешевая:

INDY R4600PC	100 МГц
Color	8-bit (Virtual 24)
Monitor	17" (1280x1024)
System RAM	32 Мбайт
Page Buffer	128 Мбайт
Media Drive	3,5" SCSI Floppy
Hard Disk	0,5 Гбайт

2) оптимальная:

INDY R4600SC	133 МГц
CHALLENGE S Server	150 МГц (дополнительный RIP-сервер)
Color	24-bit
Monitor	20" (1280x1024)
System RAM	128 Мбайт
Page Buffer	128 Мбайт
Media Drive	CD-ROM, DAT
Hard Disk	1 Гбайт (internal), 9 Гбайт (external)

Из других нетрадиционных применений станций SGI можно выделить реализованные в рамках "виртуальной реальности" программы интерактивной архитектурной визуализации. Их используют, например, в риэлтерской деятельности, чтобы клиент мог ознакомиться с планировкой помещения, прикинуть варианты расстановки мебели и т.д., не отходя от компьютера. Интересно отметить также разработку систем по производству видеоигр, интерактивного видео и других приложений.

Процессоры, системные платы и винчестеры

Что выбрать для модернизации?

Виктор Арковенко

При модернизации или сборке компьютера важно прежде всего ориентироваться на такие технические решения, которые не только останутся актуальными в течение достаточно длительного времени, но и достигли состояния определенной зрелости, которое обычно очень благоприятно сказывается на ценах. В данном случае будут рассмотрены только основные комплектующие — процессоры, системные платы и винчестеры.

Процессоры. Ситуацию на рынке процессоров определяет начало массового внедрения в производство новых технологий, обеспечивающих формирование элементов с размерами порядка 0,5 мкм (традиционные 486 процессоры и Pentium 60 фирмы Intel изготавливались по 0,8 мкм технологии). Уменьшение размера элементов обеспечивает не только повышенное быстродействие, но и помогает избежать перегрева за счет пониженного напряжения питания (3,3 В). Легко достигаются тактовые частоты 100 МГц и выше, что практически не оставляет шансов более низкочастотным изделиям. Это тем более так, что применяемый принцип умножения внутренней частоты позволяет процессорам функционировать на стандартных внешних тактовых частотах 33 или 40 МГц, не предъявляющих особых требований к остальным компонентам, то есть переход на более высокие частоты не увеличивает стоимость системной платы. Перспективность процессора определяется не только его архитектурой, но и уровнем технологии, используемой при его изготовлении. Последняя в существенной степени влияет на такое важное для потребителя соотношение, как "цена/качество". Неудивительно, поэтому, что новые все более производительные и дешевые процессоры семейства 486 являются пока наилучшим выбором для большинства пользователей (особенно в связи с проблемами с процессором Pentium, о которых будет сказано ниже). Важно отметить, что новые процессоры типа 486DX2 и 486DX4 (по крайней мере, это относится к изделиям фирмы AMD) — это одна и та же модель, имеющая переменный коэффициент внутреннего умножения частоты и допускающая использование и в качестве DX2, и в качестве DX4. Окончательная маркировка зависит, по-видимому, от условий тестирования. Поэтому можно ожидать, что по мере отработки технологии выпуск будет смещаться в сторону процессоров типа DX4, которые должны вытеснить процессоры DX2.

Для высокопроизводительных систем, совместимых с платформой Intel, в настоящее время наилучшим выбором по-прежнему остается Pentium на 90/100 МГц, (ожидается появление моделей с тактовой частотой до 150 МГц). Предпринятое фирмой Intel несколько месяцев назад существенное снижение цен на этот процессор, казалось, смогло изменить положение в его пользу и на массовом рынке. Практически ему не было серьезной альтернативы. К сожалению, обнаруженная осенью в блоке операций с плавающей точкой процессора Pentium ошибка сильно изменила ситуацию, так как, во-первых, потребовалось время на выпуск исправленных версий, и, во-вторых, значительная часть новых процессоров предназначена для замены дефектных, что не может не сказаться на ценах. Хотя Intel уверяет, что в подавляющем большинстве случаев проблем, вызванных наличием ошибки, не возникает, решать вопрос о том, продолжать ли использовать дефектный процессор, должен все-таки сам пользователь.

Несколько слов о характере ошибки. Она заключается в пониженной точности результата при выполнении операций деления с плавающей точкой для некоторых вполне определенных пар операндов. Максимальная погрешность при этом может составлять порядка $6*10^{-5}$ при вычислениях как с одинарной, так и с двойной и расширенной точностью. Для одинарной точности количество таких пар не менее 1738, причем в 87 случаях погрешность имеет близкую к максимальной величину (всего же возможно 2^{46} разных сочетаний делимого и делителя). Те же самые пары чисел дают ошибочный результат (под ошибочным здесь подразумевается результат, имеющий точность ниже ожидаемой) и при операциях деления с более высокой точностью. Вообще же "плохих" пар в этом случае несравненно больше. Оценка вероятности появления ошибки при выполнении операций деления с плавающей точкой, приводимая фирмой Intel, составляет порядка 10^{-10}, то есть совсем немного. Непонятно, правда, как все-таки быть с распространением ошибок и как определять правиль-

ность полученного результата — не пересчитывать же на 486 компьютере. Тем более, что некоторые фирмы, например IBM, не разделяют оптимистичного подхода Intel к оценке вероятности появления ошибки при использовании некоторых популярных приложений, в том числе электронных таблиц.

До того как ошибка была обнаружена, было выпущено не менее 2 миллионов процессоров Pentium, поэтому в настоящее время наряду с исправленными версиями продолжают широко предлагаться и дефектные процессоры. Отличить их очень просто. Для этого достаточно использовать пару чисел из набора "плохих" и посмотреть на результат деления. Можно воспользоваться программой "калькулятор" из Windows. Если разделить 4 195 835 на 3 145 727, то правильный ответ должен быть 1.33382045, а компьютер с процессором Pentium, содержащим ошибку, выдаст 1,33373907. Это как раз один из случаев максимальной ошибки $6*10^{-5}$, которая значительно превосходит не только уровень, соответствующий двойной точности, но и одинарной (порядка 10^{-15} и 10^{-7} соответственно). Можно также поделить 5 505 001 на 294 911. Правильный ответ будет 18,66665197, а неправильный 18,66600093. При проверке следует, однако, иметь в виду, что ошибка может быть уже скорректирована на программном уровне.

Системные платы. Изменения в процессорах, естественно, самым непосредственным образом сказались и на системных платах. Несмотря на серьезные усилия Intel по продвижению Pentium, большинство тайваньских производителей продолжали развивать также и линию 486. Можно говорить о новом поколении массовых системных плат (486 и Pentium), в которых удалось воплотить все основные особенности, свойственные ранее только изделиям Brand-Name.

- Применение шины PCI во всех моделях, кроме самого начального уровня.
- Обязательная поддержка напряжения питания 3,3 В. (Это относится как к процессорам 486DX2/DX4, так и к Pentium.)
- Практически всегда в системную плату интегрированы контроллеры винчестеров Enhanced IDE (часто с поддержкой 4-х устройств и режимов обмена mode 3 и mode 4), гибких дисков и порты (два последовательных и параллельный). Такое решение представляется наиболее оптимальным, особенно в связи с проблемами с совместимостью, о которых шла речь в предыдущей публикации. Действительно, для стандартной комплектации остается подобрать только видеоадаптер. Существуют, правда, еще более радикальные решения — это платы типа All-In-One со встроенным видеоадаптером, однако они предназначены для установки в менее распространенные корпуса типа super slim и по возможностям расширения существенно уступают традиционным платам.
- Универсальность для 486 системных плат, то есть обязательная поддержка процессоров разных производителей (Intel, AMD, Cyrix, UMC) и разных типов (SX/SX2/DX/DX2/DX4 с напряжением питания как 3,3, так и 5 В).
- В качестве оперативной памяти используются 72-контактные модули типа SIMM.
- Начинают появляться платы с поддержкой режима "plug & play", то есть с автоматическим конфигурированием.

Винчестеры. Особенности современного программного обеспечения, в том числе и из области multimedia, вывели на первый план такой параметр винчестеров, как емкость. Практически все последние годы производители стремились уменьшить время поиска и увеличить скорость вращения дисков (для увеличения скорости передачи данных и уменьшения времени доступа за счет скрытого времени). Достигнутые значения порядка 8 мс и 7200 об./мин находятся на грани технологических возможностей, что существенно сказывается на ценах. Для типовых однопользовательских применений наибольшее влияние на производительность оказывает скорость передачи данных, которую можно увеличивать не только повышением скорости вращения, но и увеличением плотности записи, что приводит также к возрастанию емкости. Именно на повышение плотности записи и направлены основные усилия производителей винчестеров. Достигнуты настолько высокие значения, что это позволило получать хорошую производительность даже при стандартных (порядка 3500-4500 об./мин) скоростях вращения, что в свою очередь позволило наладить массовый выпуск производительных, емких и очень недорогих винчестеров. Так, при применении серии 7000AV фирмы Maxtor стандарт комплектации персонального компьютера легко, практически без увеличения стоимости, может быть поднят до уровня 850 Мбайт (ожидается, что винчестеры с такой емкостью будут в этом году самыми продаваемыми). Массовые 3,5" серии всех ведущих фирм (Maxtor, Seagate, Conner, Western Digital, Quantum) имеют емкость 420-1260 Мбайт, среднее время поиска 12 мс и снабжены усовершенствованным интерфейсом IDE — Enhanced IDE (или Fast ATA, ATA-2). Этот интерфейс позволяет за счет новых режимов (mode 3 со скоростью передачи до 11,1 Мбайт/с) увеличить скорость обмена данными. Более производительные модели, например Fujitsu M1606TA имеют скорость вращения диска 5400 об./мин, среднее время поиска 10 мс и поддерживают режим mode 4 со скоростью передачи до 16,6 Мбайт/с.

При применении винчестеров с интерфейсом Enhanced IDE необходимо учитывать следующее. Свой-

ственное стандарту IDE ограничение емкости величиной 504 Мбайт преодолевается путем использования для адресации логических блоков (LBA — Logic Block Address) вместо указания номера головки, цилиндра и сектора. Поэтому необходимо, чтобы BIOS системной платы поддерживал режим LBA. Для старых версий BIOS требуется специальный драйвер. Такой драйвер поставляется фирмой Maxtor для моделей 7540AV, 7850AV, 71050A, 71260A (емкостью 540, 850, 1050, 1260 Мбайт) в составе специального пакета MaxBlast (Maxtor — Ontrack Disk Manager). Этот пакет содержит также драйверы, позволяющие преодолеть даже такое типичное для новых винчестеров емкостью более 504 Мбайт затруднение, как поддержка 32-битного доступа к диску под Windows.

Практические замечания. Поскольку номенклатура комплектующих очень широка, и они не всегда равноценны по качеству, следует обратить внимание и на некоторые детали чисто практического плана.

- Появившиеся сейчас процессоры фирмы AMD 486DX2-66/80 с напряжением питания 3,3 В при установке в некоторые системные платы, сконфигурированные под DX2-66/80, опознаются как DX4-100/120 и при этом показывают соответствующую производительность. То есть процессор работает не с удвоением, а с утроением внутренней частоты. Это связано с особенностями как процессоров (о них упоминалось ранее), так и системных плат, которые, вообще говоря, не рассчитаны на установку таких процессоров. Вопрос о целесообразности применения процессоров в этом режиме, то есть при повышенных тактовых частотах, остается открытым. Несмотря на наличие некоторого положительного опыта, не исключено снижение надежности и даже времени службы.
- При выборе системных плат следует обращать внимание на реальную поддержку напряжения питания 3,3 В (для этого на плате должен быть смонтирован соответствующий преобразователь). Приобрести его и установить самостоятельно довольно сложно. Кроме того, следует иметь в виду, что для некоторых экземпляров процессоров для нормальной работы требуется несколько более высокое напряжение, чем 3,3 В. Именно поэтому уровень напряжения на многих платах слегка повышен.
- К сожалению, жесткая конкуренция подталкивает некоторых производителей и продавцов дешевой продукции к не вполне корректному поведению (для последних, зачастую, помимо их воли). Так, предлагаются иногда системные платы со следующими неприятными особенностями. В некоторых из них, с запаянными микросхемами кэш-памяти, на самом деле может быть установлено не 256, а только 128 Кбайт, остальные 128 Кбайт — это просто муляжи. Система всегда будет показывать наличие 256 Кбайт, поэтому отличить такие платы можно только по понижению производительности при выполнении некоторых чувствительных к эффективности кэширования приложений. С целью уменьшения стоимости применяются очень тонкий стеклотекстолит и разъемы слотов с некачественным покрытием, что резко увеличивает опасность механических повреждений и снижает надежность контактов.
- Часто модули памяти типа SIMM с эмуляцией контроля четности выдаются за модули с истинным контролем четности, так как логические микросхемы в них перемаркированы под микросхемы динамической памяти. Было бы разумнее, поэтому, применять модули вообще без контроля четности, которые к тому же заметно дешевле, так как переплачивать за совершенно бесполезную эмуляцию нет никакого смысла.

В очередном занятии «Курса молодого бойца» мы обсудим проблемы профилактики и контроля состояния дисковых данных, а также начнем описание способов восстановления поврежденной или утерянной информации.

Курс молодого бойца

ЗАНЯТИЕ ПЯТНАДЦАТОЕ

Камилл Ахметов

▲CHKDSK

Об этой старой, как DOS, программе все еще не стоит забывать. Мы начнем изучение средств «техобслуживания» дисков с этой утилиты не только для того, чтобы проследить эволюцию дисковых утилит с самого начала. Наиболее часто встречающейся ошибочной ситуацией является появление «потерянных кластеров», и CHKDSK всегда прекрасно справляется с ней. Кроме того, эта программа DOS всегда является наиболее быстрым способом определить, все ли в порядке на диске, а также выяснить объем доступной базовой памяти DOS и свободного пространства на диске.

Для начала вспомним синтаксис программы CHKDSK:

```
CHKDSK  [диск:][путь][имяфайла][/F][/V]
```

Если задано имя файла (допустимо использование шаблонов), CHKDSK будет проверять диск на степень фрагментирования и отображать отчет о состоянии файла. С ключом **/F** программа попытается устранить ошибки в каталоге или в таблице размещения файлов, с ключом **/V** — отобразит все файлы и их полные имена на указанном диске. Позволю себе заметить, что если у вас есть книга «Курс молодого бойца», то у вас есть также и вся необходимая справочная информация по основным командам DOS.

Испытаем программу CHKDSK в действии — проверим состояние 1,44-мегабайтной дискеты, на которой находится шесть файлов.

```
Volume Serial Number is 3B62-07C2

    1 457 664 bytes total disk space
      308 736 bytes in 6 user files
    1 148 928 bytes available on disk

          512 bytes in each allocation unit
        2 847 total allocation units on disk
        2 244 available allocation units on disk

      655 360 total bytes memory
      607 120 bytes free
```

Обратите внимание на то, что под «единицей размещения» (allocation unit) в этом сообщении понимается кластер, а не сектор (хотя на 1,44-мегабайтных дискетах кластер состоит из одного сектора). Ну и, кроме того, мы получили информацию о состоянии базовой памяти DOS.

А теперь — «правдивая история из жизни». В результате некоторых операций с этой дискетой три файла из шести должны были быть удалены. Что-то — сбой в программе или аппаратная проблема — помешало корректно провести удаление файлов. Как известно, процедура удаления файла в DOS состоит из замены первого символа имени файла в записи каталога на символ с кодом ASCII 229 (то есть «s», в 866-й кодовой странице — «х») и записи во все поля таблицы размещения файлов (File Allocation Table, FAT), соответствовавшие кластерам этого файла, значения 000h (или 0000h, в зависимости от формата FAT), что значит «кластер свободен». Допустим, эта операция не была завершена, и на диске оказались кластеры, занятые согласно таблице FAT, но не принадлежащие ни одному файлу. Именно такие кластеры и называются «потерянными».

Потерянные кластеры могут возникать на дисках, с которыми вы работаете, чаще, чем вы это можете предположить. Они могут даже не мешать работе программ, поскольку DOS при записи новых файлов на диск «обходит» потерянные участки точно так же, как и занятые. Пока вы сами не позаботитесь о том, чтобы проконтролировать порядок на диске, никто вам не укажет на потерянные кластеры.

Теперь посмотрим, что сообщит CHKDSK после проверки той же дискеты, на которой потерялись три файла:

```
Volume Serial Number is 3B62-07C2
Errors found, F parameter not specified
Corrections will not be written to disk

   32 lost allocation units found in 3 chains.
       16 384 bytes disk space would be freed

    1 457 664 bytes total disk space
      292 352 bytes in 3 user files
    1 148 928 bytes available on disk

          512 bytes in each allocation unit
        2 847 total allocation units on disk
        2 244 available allocation units on disk

      655 360 total bytes memory
      607 120 bytes free
```

CHKDSK подсказывает, что положение на дискете можно исправить, загрузив CHKDSK с параметром

«/F». Если так и поступить, то прежде чем производить какие-то изменения на диске, CHKDSK спросит:

```
Convert lost chains to files (Y/N)?
```

Отрицательный ответ приведет к тому, что CHKDSK просто «вычистит» FAT и освободит таким образом все потерянное пространство. Если же ответить «Y», то в корневом каталоге диска появятся файлы с именами вида «FILE????.CHK», по количеству цепочек потерянных кластеров. В нашем случае появится не менее трех файлов: FILE000.CHK, FILE0001.CHK и FILE0002.CHK, а может и больше — если порядок потерянных цепочек нарушен.

В любом случае CHK-файлы не будут идентичны потерянным файлам, поскольку они всегда состоят из целого количества кластеров, без «хвостов». Поэтому таким образом имеет смысл восстанавливать текстовую информацию, но не файлы сложных форматов, которые нельзя прочитать в случае нарушения структуры файла (файлы электронных таблиц, баз данных и тому подобные). Откровенно говоря, на запрос о конвертировании потерянных цепочек в файлы практически всегда можно отвечать отрицательно.

Могут быть встречены дефекты другого рода. Некоторые из них также бывают связаны с несовершенством системы каталогов DOS и таблицы FAT. В результате сбоев, аналогичных описанному, могут появиться ошибки в данных каталогов о размерах файлов или размещении их первых кластеров, такие ошибки относятся к ошибкам размещения (allocation error). В таблицу FAT может попасть более одной ссылки на один и тот же кластер диска, тогда получится, что этот кластер принадлежит сразу двум файлам, такие файлы называются перекрестными (cross linked). Сообщения программы CHKDSK о таких ошибках выглядят следующим образом:

```
A:\DOC1.DOC
    Allocation error, size adjusted
A:\DOC2.DOC
    Allocation error, size adjusted
A:\GLAVA3.WRI
    Allocation error, size adjusted
A:\DOC2.DOC
    Is cross linked on allocation unit 799
A:\GLAVA5.WRI
    Is cross linked on allocation unit 799
```

Доверять программе CHKDSK исправление ошибок размещения или перекрестных файлов нельзя. При отсутствии других инструментов можно попробовать выйти из положения следующим образом — скопировать файлы, в которых обнаружены такие неисправности, на надежный диск (или флоппи-диск) и удалить оригиналы. Предпочтительнее, конечно, доверить коррекцию таких ошибок более развитым программам, чем CHKDSK, — например Microsoft ScanDisk.

▲ScanDisk

В экранных сообщениях программы CHKDSK, начиная с версии MS-DOS 6.2 вы найдете следующую фразу: «Попробуйте вместо CHKDSK использовать ScanDisk. ScanDisk достоверно обнаруживает и исправляет больше разновидностей дисковых ошибок». Это действительно так, и если вы работаете с MS-DOS 6.2x, то вам следует полагаться именно на ScanDisk (рис. 1).

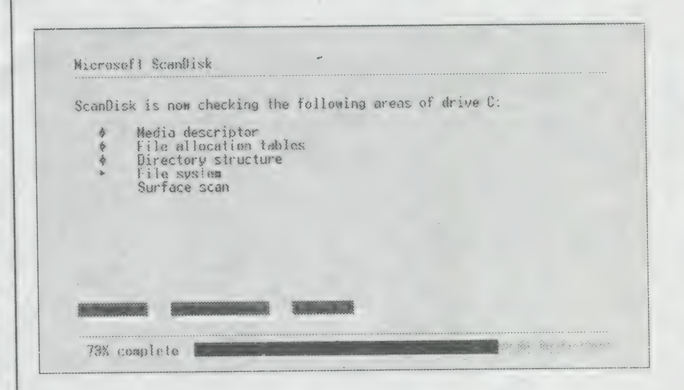


Рис. 1

Кроме того, что ScanDisk с честью выходит из значительно более сложных ситуаций, чем потеря кластеров, она обнаруживает на диске физически неисправные кластеры и пытается перенести данные, находящиеся в них, на другие участки диска, а дефектные области помечает как непригодные для дальнейшего использования. По окончании работы ScanDisk выводит на экран отчет с перечислением обнаруженных ошибок и сделанных изменений (рис. 2).

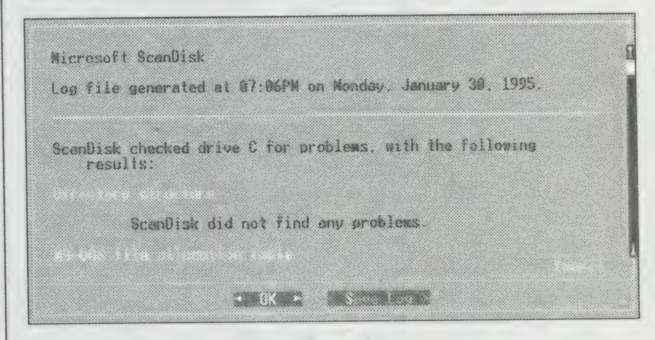


Рис. 2

По умолчанию сеанс работы ScanDisk выглядит следующим образом — программа сразу начинает проверять логическую структуру текущего диска и выдает запрос на подтверждение исправления каждой ошибки. Для страховки можно сохранить на другом диске информацию о состоянии дисковых данных до исправления ошибок. Затем ScanDisk спра-

шивает, нужно ли производить тест поверхности диска, и в случае физических ошибок на диске также запрашивает подтверждение на коррекцию. После всех перечисленных действий ScanDisk предлагает пользователю прочитать отчет, и на этом сеанс работы заканчивается.

Правильным решением будет поместить команду вызова такой полезной программы в AUTOEXEC.BAT. Но в пакетном режиме программа должна работать без «разговоров». Для того чтобы настроить ScanDisk на работу без запросов, существует несколько ключей запуска программы.

ScanDisk будет проверять либо текущий диск, либо диск, буква которого указана в параметрах команды. Ключ **«/ALL»** заставит программу проверять все жесткие диски компьютера. Можно ошибки исправлять автоматически, для этого служит параметр **«/AUTOFIX»** (и **«/NOSAVE»** — для пропуска стадии сохранения страховочной информации). Чтобы программа не останавливалась для выдачи отчета, применяется ключ **«/NOSUMMARY»**. Если в режиме автоматической коррекции ошибок задать **«/SURFACE»**, то программа проделает тест поверхности, а если нет — то без лишних вопросов опустит его.

Таким образом, в файл AUTOEXEC.BAT можно смело поместить команду

```
SCANDISK /ALL /AUTOFIX /NOSAVE /NOSUMMARY
```

Если же вы хотите, чтобы при каждой загрузке производился тест поверхности диска, команда должна выглядеть следующим образом:

```
SCANDISK /ALL /AUTOFIX /NOSAVE /SURFACE /NOSUMMARY
```

Прервать работу программы ScanDisk очень просто — нажатием клавиши Esc. По окончании пакетного сеанса работы ScanDisk оставляет в корневом каталоге диска файл SCANDISK.LOG с отчетом.

Как и CHKDSK, ScanDisk может проверять степень фрагментации файлов. Это делается по команде вида

```
SCANDISK имяфайла /FRAGMENT
```

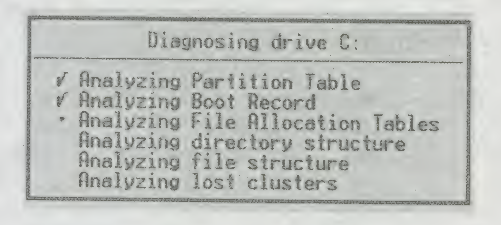
Отчет о фрагментации может выглядеть так:

```
C:\DOC\WORK\WORD_9.DSK contains 16 noncontiguous blocks.
C:\DOC\WORK\~WRA0003.ASD contains 2 noncontiguous blocks.
C:\DOC\WORK\~WRA2995.ASD contains 2 noncontiguous blocks.
```

▲ Norton Disk Doctor

В принципе, Microsoft ScanDisk является вполне приемлемой программой. Однако не зря существуют и прекрасно продаются специализированные пакеты дисковых утилит. Программа Norton Disk Doctor (рис. 3) из пакета Norton Utilities 8.0 фирмы Symantec предоставляет пользователю больше возможностей управления процессом диагностики и лечения дисковых ошибок.

```
ndd c: /q
```




```
Analyzing File Allocation Tables...
2nd FAT, 73rd sector
Sectors Processed: 325          64%
```

Рис. 3

Для запуска NDD из командной строки достаточно знать три варианта ключа для пакетной работы: **«/C»** для полной проверки диска, **«/Q»** для проверки без теста поверхности диска и **«/DT»** только для теста поверхности. Таким образом, для того чтобы произвести полную проверку текущего диска, следует дать команду

```
NDD /C
```

Если в системе более одного диска и необходимо проверить их все, то следует перечислить их в параметрах команды, например, так:

```
NDD C: D: E: /C
```

Существуют и другие ключи запуска NDD для более тонкой настройки программы, но мы не будем их здесь приводить. Во-первых, о них легко узнать по команде

```
NDD /?
```

Во-вторых, еще легче настроить NDD для работы в необходимом вам режиме, воспользовавшись для этого меню самой программы. Если загрузить NDD без параметров, то на экране появится главное меню программы (рис. 4), из которого следует выбрать пункт «Options» для настройки параметров NDD. Другие пункты главного меню NDD означают следующее:
- «Diagnose Disk» — полный цикл тестирования диска;
- «Surface Test» — только тест физической поверхности диска (рис. 5), без возможности внесения исправлений;



- «Undo Changes» — отмена изменений, сделанных NDD на диск. Она возможна, если в ходе исправления ошибок была создана страховочная дискета (Undo), а дисковые данные не редактировались.

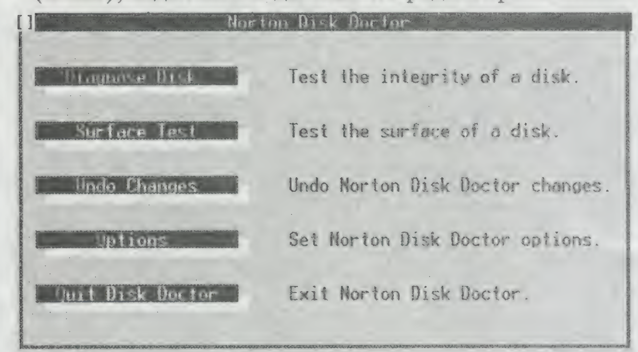


Рис. 4

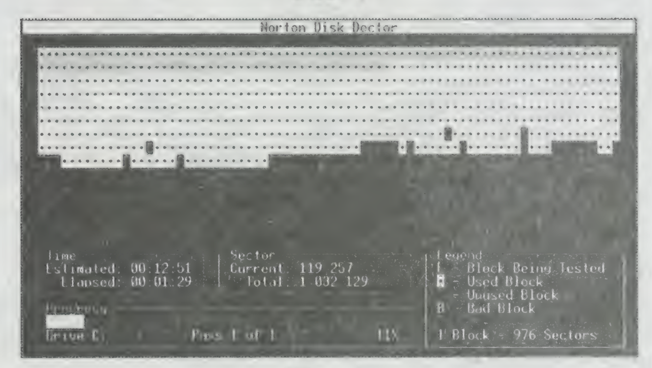


Рис. 5

Появится меню «Norton Disk Doctor Options» (рис. 6), в котором нас прежде всего интересуют пункты «General», «Surface Test» и «Auto-Repair». Остальные пункты этого меню тривиальны. «Custom Message» позволяет отконфигурировать NDD на выдачу шаблонного сообщения об ошибке типа «**Серьезная ошибка диска! Позвоните системному инженеру Вашего отдела и ничего не предпринимайте до его прихода**» для неквалифицированных пользователей. «Save» сохранит отредактированные установки для последующих сеансов работы NDD, «OK» — только для текущего сеанса, «Cancel» отменит изменения.

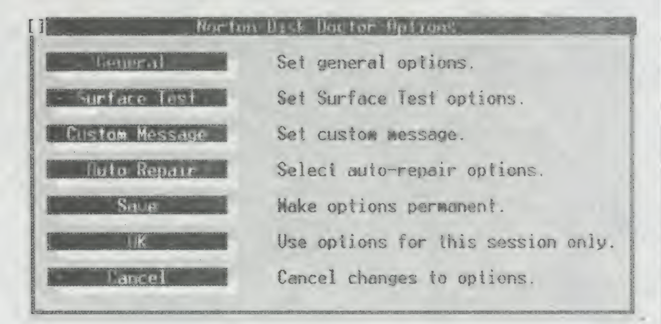


Рис. 6

Меню «General Options» (рис. 7) позволит задать NDD основные параметры, а именно:

- Как выполнять тест поверхности диска:
- с предварительным запросом;
- вообще не тестировать поверхность;
- тестировать поверхность без подтверждения.
- Выполнять или пропускать следующие тесты:
- тест таблицы разбиения жесткого диска;
- тест корректности установок CMOS (параметров настройки оборудования, сохраняемых в энергонезависимой схеме Complementary Metal-Oxide Semiconductor); этот тест следует отменить, если на компьютере нестандартная схема CMOS;
- тест корректности данных на диске, сжатом программой динамической компрессии данных; этот тест придется отменить, если NDD не поддерживает используемую программу сжатия, например, Stacker 4.0;
- тест несжатого диска, на котором находятся сжатые;
- тест ТОЛЬКО жесткого диска номер 1, в тех случаях, когда NDD «кажется», будто на вашем компьютере больше одного винчестера, а на самом деле один.
- Какие диски не следует проверять.

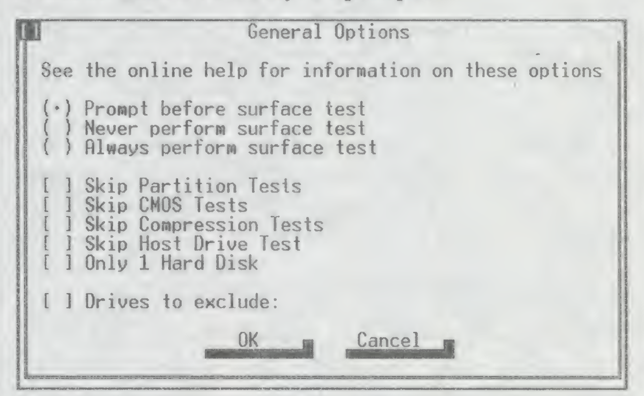


Рис. 7

Меню «Surface Test Options» (рис. 8) позволяет установить по своему вкусу настройки теста поверхности.

Если требуется проверять именно состояние дисковых данных, а не всей поверхности диска, то можно сэкономить массу времени (тест поверхности — достаточно долгий процесс), проверяя только ту площадь диска, которая занята данными («Area Used by Files»). NDD обеспечивает два режима проверки физической поверхности диска — стандартный («Normal Test»), проверяющий данные на уровне дорожек, и более долгий и тщательный («Thorough Test»), тестирующий каждый сектор. Интересная возможность NDD — осуществлять полное тестирование только по пятницам, словно генеральную уборку. Наконец, можно задать NDD количество повторений («Repetitions») проверки или заставить программу гонять тест «по кругу», пока не поступит команда от пользователя.

```
                Surface Test Options

            Select the default settings
              for the Surface Test.

  What to Test
  (•) Entire Disk Area
  ( ) Area Used by Files

  Type of Test
  ( ) Normal Test
  ( ) Thorough Test
  (•) Thorough Test only on Fridays

  Passes
  (•) Repetitions [1..]
  ( ) Continuous

             OK          Cancel
```

Рис. 8

Самый интересный пункт меню NDD — «Automatic Repair Options» (рис. 9). NDD можно настроить на автоматическую коррекцию ошибки без участия пользователя («Auto»). По умолчанию NDD запрашивает подтверждение пользователя на исправление любой ошибки («Prompt»). Есть и третья возможность — пропускать найденные ошибки без запросов и без коррекции. Это нужно, когда NDD загружается только для диагностики. Кроме того, это помогает в тех (чрезвычайно редких) случаях, когда NDD находит ошибки, которых нет — например с неподдерживаемой системой динамического сжатия данных.

```
                     Automatic Repair Options

  Partition Tables:          ( ) Prompt  (•) Auto  ( ) Never
  Boot Records:              ( ) Prompt  (•) Auto  ( ) Never
  File Allocation Tables:    ( ) Prompt  (•) Auto  ( ) Never
  Directory Structures:      ( ) Prompt  (•) Auto  ( ) Never
  Compression Structures:    ( ) Prompt  (•) Auto  ( ) Never
  Surface Errors:            ( ) Prompt  (•) Auto  ( ) Never

  [x] Auto-repair lost clusters if fewer than [25..] chains

  ( ) Save lost clusters as files
  (•) Delete lost clusters

  ( ) Prompt for creation of Undo file
  (•) Make repairs without creating Undo file

                         OK        Cancel
```

Рис. 9

Таким образом можно регулировать режим исправления ошибок для:
- таблицы разбиения физического диска;
- загрузочных записей логических дисков и дискет;
- таблицы размещения файлов;
- каталогов;
- сжатых дисков;
- поверхности дисков.

В этом же меню можно задать поведение NDD по отношению к потерянным кластерам, а также приказать программе запрашивать (или не запрашивать) дискету для сохранения страховочных данных.

Как мы видим, запуск полностью настроенного для работы в пакетном режиме NDD из AUTOEXEC.BAT командой «NDD /Q» — еще более удачный вариант, чем использование Microsoft ScanDisk. Еще одно отличие NDD заключается в том, что если в пакетном сеансе работы NDD произвел какие-либо изменения на диске, то он обязательно предлагает пользователю ознакомиться с ними — просмотреть отчет. Можно настроить программу на постоянную выдачу отчета в файл ключом **«/R:имяфайла»**, а с ключом **«/RA:имяфайла»** — на добавление каждого нового отчета к файлу с предыдущими.

С пакетом Norton Utilities 8.0 поставляется программа NDD для Windows. Она имеет несколько другой интерфейс (рис. 10). Все пункты настройки NDDW, касающиеся пропускаемых тестов, теста поверхности и автоматической коррекции, совпадают с такими же пунктами NDD.

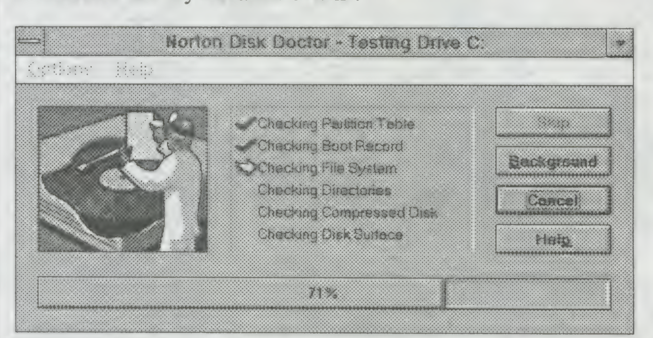


Рис. 10

Практически все операции тестирования программой NDDW идут медленнее, чем аналогичные операции программы NDD. Ценность NDDW, однако, заключается в том, что эта программа может работать — тестировать дисковые данные и исправлять ошибки — в фоновом режиме. На практике это происходит так — если NDDW отконфигурирован для работы в фоновом режиме, то он начинает работу, например, спустя 15 секунд после того, как на компьютере не производилось никаких действий, и останавливает работу, если пользователь нажимает клавишу или двигает мышью.

NDD для DOS обладает способностью восстанавливать разрушенные таблицы разбиения жестких дисков по найденной копии. Эта операция производится по команде «NDD /REBUILD». Копию системной области следует делать при помощи таких утилит, как программа Image из Norton Utilities, или Mirror из PC Tools или MS-DOS 5.0.

(Продолжение следует)

Очередное занятие по программированию, которое ведет преподаватель Лицея Информационных Технологий М.Антóнова, посвящено началам использования языка С.

Начинаем работать на С

Марианна Антонова

В 10 классе Лицея Информационных Технологий лицеисты, выбравшие специальность "Программист", изучают, как вы уже знаете, виртуальную машину ММ, ее систему команд. К концу первого семестра 10 класса они понимают, что такое плавающее и фиксированное число, как организуются циклы, как обращаться к подпрограммам, — все это на уровне архитектуры виртуальной машины. Они успевают отладить, используя модель ММ на PC, десяток программ.

Программы эти просты, но содержат все необходимые компоненты: программирование формул, работа с битами (логическое сложение, умножение, сдвиг), организация циклов, работа с массивами чисел, обращение к подпрограммам. При работе с моделью ММ на PC ученики используют оболочку, подобную среде Turbo C, то есть имеют возможность отлаживать программы по шагам, останавливать выполнение программ по необходимому адресу, просматривать значения всех переменных в момент останова.

Все это дает возможность быстро перейти к программированию на С, используя среду Turbo C. Обучение построено таким образом, чтобы можно было быстро писать и отлаживать программы на С. Поэтому сразу начинаем с такой задачи:

▲Задача о флаге

Пусть дан массив из N чисел, содержащий только 0, 1, 2. Требуется нарисовать прямоугольник тремя цветами так, чтобы ширина прямоугольника каждого цвета соответствовала процентному отношению нулей, единиц и двоек в исходном массиве. При этом три цвета нужно выбрать из палитры — набора 16 цветов — с помощью курсора. Форма курсора произвольна. Прямоугольник для флага требуется расположить на нижних 2/3 экрана. В верхней трети экрана нужно вывести прямоугольник палитры с 16 цветами.

Каким образом можно написать и отладить такую программу? Во-первых, разобьем ее на логические части. Необходимо определить, что именно должна выполнить каждая часть и какая информация для этого нужна.

Можно, например, выделить следующие логические части:

Логическая часть	Входная информация	Выходная информация
Ввод исходного массива	Массив 0, 1, 2	Массив 0, 1, 2 в памяти
Вычисление процентного отношения нулей, единиц и двоек в массиве	Массив 0, 1, 2 в памяти	Процентные отношения 0, 1, 2
Вывод на экран прямоугольного массива из 16 цветов	Нет	Прямоугольный цветовой массив шириной 50 пикселов. Длина каждого прямоугольника 40 пикселов
Выбор трех цветов из набора из 16 цветов	Прямоугольный цветовой массив на экране	3 цвета (их номера)
Вычисление ширины прямоугольника каждого цвета	Процентное отношение нулей, единиц и двоек в исходном массиве, ширина всего прямоугольника	Ширина каждого из 3 прямоугольников (т.е. координаты верхнего левого и правого нижнего угла каждого прямоугольника)
Рисование прямоугольного массива в соответствии с выбранными цветами и шириной каждого прямоугольника	Номера 3-х выбранных цветов и координаты верхнего левого и правого нижнего угла каждого прямоугольника	Картинка на экране

Расписав таким образом этапы, приступаем к написанию программы.

Каждый отдельный этап будет у нас оформлен в отдельную функцию без параметров, которая пока будет только печатать свое название. Такую программу на С написать очень просто.

```
/* Программа печатает название функций */
```

— так оформляется комментарий на С, то есть комментарий начинается с /* и заканчивается */

```
#include <stdio.h>

/* файл stdio.h содержит информацию для функции ввода-вывода. Мы подключаем
        эту информацию к своей программе */

main()  /* main - имя главной функции, с которой начинается выполнение
        программы. Пустые скобки означают, что main не имеет параметров */
{       /* фигурная скобка открывает тело функции */
        input_mas();        /* обращение к функции ввода исходного массива.
                            После ее выполнения мы возвращаемся в главную
                            программу */
        percent();          /* обращение к функции вычисления процентного
                            отношения */
        color_box();        /* обращение к функции вывода цветного
                            прямоугольника из 16 цветов */
        color_3();          /* выбор трех цветов */
        width_3();          /* вычисление ширины каждого прямоугольника */
        draw_box();         /* рисование прямоугольника выбранными цветами */
}       /* скобка закрывает main */
```

Далее идут сами функции (они могут быть написаны и до main):

```
input_mas()
{
        printf("input_mas\n");      /* печатаем название функции; \n
                                    переводит строку после печати */

}       /* скобка закрывает тело функции */

percent()
{
        printf("percent\n");
}

color_box()
{
        printf("color_box\n");
}

color_3()
{
        printf("color_3\n");
}

width_3()
{
        printf("width_3\n");
}

draw_box()
{
        printf("draw_box\n");
}
```

Теперь можно отладить эту программу в среде Turbo C. Следует научиться редактировать программу, осуществлять автоматический запуск или работать по шагам. Кроме того, необходимо научиться также останавливаться на выбранной строке и смотреть значения переменных.

Теперь мы будем постепенно "заполнять" отдельные функции. Начнем с input_mas.

```
input_mas()
{
        int i;
        printf("input_mas\n");      /* печать названия самой функции */
        printf("Введите исходный массив\n");
        for ( i=0; i<N; i++)        /* цикл по i от 0 до N-1 с шагом 1,
                                    тело цикла работает N раз */

                scanf("%d",&MAS[i]);
}
```

В этой функции есть переменные, которые не определялись и под которые, следовательно, память отведена не была. Прежде чем определять переменные, следует решить — в какой точке программы отводить под них память? Если это сделать внутри функции, то переменными, определенными таким образом, нельзя будет воспользоваться в других функциях. Все переменные, определенные внутри функции, «заключены» в ней — это локальные переменные.

Если же определить массив вне функции, но до нее, то элементами массива можно будет пользоваться всюду от места определения до конца программы. Но чтобы определить массив, необходимо знать, сколько в нем элементов.

Пусть в массиве N элементов. Тогда описание массива будет таким:

```
int MAS[N],
```

где N — константа, ранее определенная с помощью

```
#define N 20        /* определим N как число 20. Всюду, где в программе
                    встретится N, оно будет заменено на число 20. Память под N
                    не отводится: это замена по написанию.*/
```

Теперь обсудим цикл. В указанной функции это:

```
for ( i=0; i<N; i++ ) scanf("%d",&MAS[i]);
```

Этот цикл работает по схеме, изображенной на рис. 1.

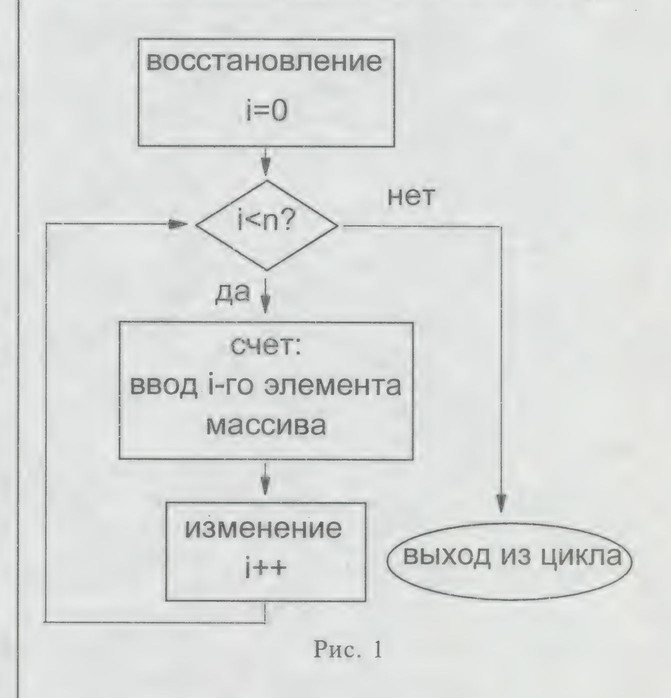
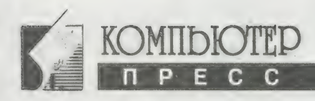


Рис. 1

С помощью функции scanf("%d",&MAS[i]) мы читаем нули, единицы или двойки из буфера клавиатуры в массив с именем MAS, заполняя MAS[0],



MAS[1] и т.д. до MAS[N-1]. На вводе ожидаются целые числа, о чем говорит ключ формата %d.

Теперь напишем функцию percent

```
percent()
{
        int i, sh0, sh1, sh2;        /* отвели память под целые переменные
                                     i, счетчик нулей sh0, счетчик единиц
                                     sh1, счетчик двоек sh2 */
        sh0=sh1=sh2=0;     /* обнулили все счетчики */

        for (i=0;i<N;i++)  /* цикл по массиву */
        {
                if (MAS[i]==1) sh1++;        /* увеличение sh1 на 1, если
                                             элемент массива равен 1 */
                if (MAS[i]==2 sh2++;
        }
        sh0=N-sh1-sh2;
        psh0=sh0*100/N;
        psh1=sh1*100/N;
        psh2=sh2*100/N;
}                          /*конец функции*/
```

Здесь уместно отметить разницу между MAS[i]=1 и MAS[i]==1. В первом случае в элемент массива заносится число 1, а во втором результатом операции будет 0 или 1 (результат примет значение 0, если элемент MAS[i] не равен 1, и 1, если MAS[i] равен 1).

В операторе if (MAS[i]==1) sh1++; sh1 будет увеличен на 1, если результат операции сравнения MAS[i]==1 равен 1 (истина), и не будет увеличен, если результат равен 0 (ложь).

Вообще говоря, мы можем использовать условные операторы двух типов:

1)

```
if (V) { оператор; оператор; ...}        /* фигурные скобки нужны, если при
                                         истинности V нужно выполнить несколько
                                         операторов */
```

V — любое выражение



Рис. 2

2)

```
if (V) { оператор; оператор; ...}
        else  { оператор; оператор; ...}
```

В первом случае блок-схема выполнения такая, как на рис. 2.

Блок-схема для второго случая изображена на рис. 3.

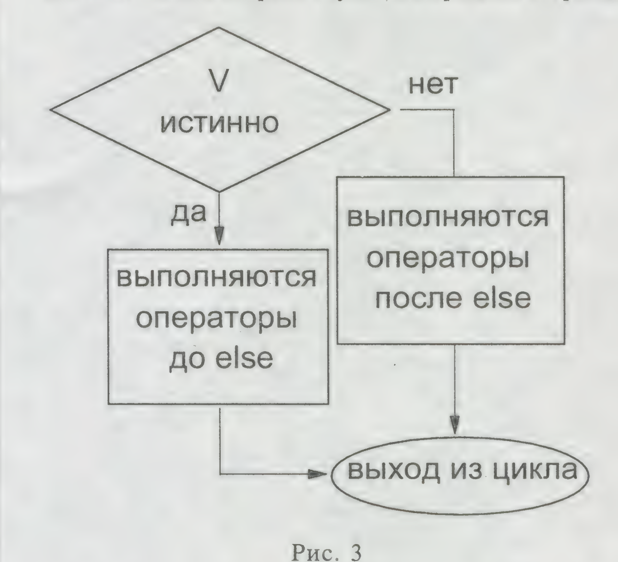


Рис. 3

Напишем функцию color_box.

```
#include <graphics.h>        /* в файле graphics.h лежит информация,
                             необходимая для графической библиотеки TURBO C,
                             которой мы будем пользоваться */

color_box()
{ int i;                     /* отводим память под переменную i, которая будет
                             использована как переменная цикла */
int a=DETECT,b;              /* эту и следующую строчку оставим пока без
                             объяснения */
initgraph( &a,&b,"e:\\tc\\bgi");
for (i=0; i<16; i++)
{       setfillstyle(SOLID_FILL,i);
        bar(40*i,0,40*(i+1),50);     /* нарисовать прямоугольник длиной 40 и
                                     шириной 50, цвет заливки i */
    }
}
```

Написанную программу следует отладить.

Прежде чем переходить к написанию функции выбора трех цветов с помощью курсора, познакомимся с кодами клавиш, в частности клавиш-стрелок. Курсор должен передвигаться, когда на клавиатуре нажимается одна из клавиш управления курсором. Как опознать, какая именно клавиша-стрелка нажата?

Отложим программу, которую мы пишем, и попробуем написать маленькую программу печати кодов клавиатуры. При нажатии клавиш выдаются так называемые ASCII- и SCAN-коды клавиш. Для функциональной клавиатуры (например, стрелок) ASCII-код равен 0, а SCAN-коды различаются, они уникальны для каждой клавиши и являются в некотором смысле ее координатами на клавиатуре.



Читать SCAN- и ASCII-коды можно, используя функцию bioskey. Для того чтобы воспользоваться этой функцией, к программе нужно подключить файл bios.h.

```
void main(void)
{
        int c=0;  /* сюда мы будем читать и SCAN-, и ACSII-коды. SCAN в
                  старший байт c, а ASCII в младший */

        char c1;  /* в c1 будет ASCII-код */
        char c2;  /* в c2 будет SCAN-код */
        while (c!=238)      /* цикл будет работать, пока не нажмем клавишу
                            ESC, имеющую код 238 */
        {
                  c=bioskey(0);       /* прочитали SCAN- и ASCII-коды*/
                  c1=c&0x00ff;        /* в c1 получили ASCII-код вырезанием
                                      младшего байта */
                  c2=c>>8;            /* получили SCAN-код */
                  printf("%d             %d\n",c1,c2);
                                      /* печатаем SCAN- и ASCII-коды */
        }
}
```

Что нового появилось в этой программе? Во-первых, цикл while.

Его общий вид:

```
While (V) { тело цикла }
```

Выполняется он так, как показано на блок-схеме на рис. 4.

В нашем случае проверяется условие **«c!=283»** (!= означает "не равно"). 283 — это общий код клавиши ESC. Эта клавиша имеет SCAN-код, равный 1, и ASCII-код, равный 27. 1 стоит в старшем байте **c**, а 27 в младшем. Следовательно, это число



Рис. 4

```
0000000100011011
```

то есть 283.

Другое незнакомое нам место:

```
c1=c&0x00ff;
```

Символ «&» — это знак побитового логического умножения. Такая операция используется, когда необходимо выделить часть ячейки; в данном случае нужен младший байт (то есть код ASCII). Поэтому мы логически умножаем **c** на константу 0x00ff (форма «0x» говорит о том, что константа шестнадцатиричная). Эта константа имет вид

```
0000000111z11111
```

Константа содержит все единицы в младшем байте и, следовательно, может служить «извлекателем» младшего байта. А что такое **«c1=c>>8»**? Знаки «>>» служат для обозначения операции сдвига вправо. Указанный оператор сдвигает величину в ячейке **c** вправо на 8 разрядов для получения SCAN-кода (вы помните, что SCAN-код стоит в старшем байте ячейки **c**?).

Отладив эту программу, мы можем получить (не заглядывая в таблицы) коды клавиш — в частности, коды стрелок, нужных нам для написания функции перемещения курсора.

Сделаем так: пусть стрелка «б» называется UP; «в» — DOWN; «а» — RIGHT; «Я» — LEFT. Напишем файл с именем KEY.H, в котором будет следующее:

```
#define  UP     72
#define  DOWN   80
#define  RIGHT  77
#define  LEFT   75
```

Эти коды мы знаем, поскольку написали программу печати SCAN- и ASCII-кодов клавиатуры. Присоединим файл KEY.H к программе, написав перед функцией

```
#include "key.h"
```

Вот теперь уже можно писать функцию выбора трех цветов с помощью курсора.

(Продолжение следует)

Преподаватели Лицея Информационных Технологий продолжают цикл статей по материалам учебного курса «Компьютерная графика». В рамках этого курса учащиеся ЛИТ осваивают материал таких спецкурсов, как «Взаимодействие Человек—компьютер», «Основы работы на персональных компьютерах», «Методы работы с графическими объектами». В этом спектре стартовых спецкурсов даются основные знания о представлении цвета в персональном компьютере, вводятся понятия палитры и графических форматов данных. Для получения качественных графических изображений — слайдов для показа на экране монитора и распечаток на принтере — необходимо знание таких понятий, как палитра и графические форматы файлов.

Фундамент всего — графические объекты

Наталья Платонова

Основным графическим устройством, с которым чаще всего приходится работать, является видеосистема компьютера. Обычно она состоит из видеокарты (адаптера) и подключенного к ней монитора. Изображение хранится в растровом виде в памяти видеокарты: аппаратура карты обеспечивает регулярное (50 и более раз в секунду) чтение видеопамяти и ее отображение на экране монитора.

Любой цветной монитор получает цвета путем наложения трех базовых цветов — красного, зеленого и синего — разных интенсивностей. Максимальная интенсивность всех трех цветов приведет к получению белого цвета, минимальная, то есть нулевая, — естественно, черного.

При записи файла изображения, использующего монохромную палитру, необходим 1 bpp (бит на точку, или бит на пиксел), значение бита 0 или 1 означает белый или черный цвет точки. При 1 байт (8 бит) на пиксел, можно отобразить 28=256 цветов, такой формат изображения служит для палитр в 64 и 256 цветов. Для 8 и 16 цветов достаточно 4 бита. В издательских системах применяются формат TrueColor (3 байта на точку, 16 777 216 цветов). Для каждого первичного цвета в формате TrueColor используется 8 бит на точку, что соответствует 256 полутонам для красного, зеленого и синего (256*256*256=16 777 216).

Никогда раньше не предоставлялось таких обширных возможностей для создания цветных изображений на персональном компьютере, как сейчас, когда вы можете иметь в своем распоряжении палитру из 16,8 млн. цветов. Такое разнообразие должно удовлетворить всех. И все же 16,8 млн. цветов — это намного меньше того количества, которое воспринимается глазом. Большинство людей может различать от 16 до 64 оттенков каждого из основных цветов. Точное количество различаемых оттенков зависит от конкретного человека, изображения и внешнего освещения.

▲Цветовые модели

На практике используются четыре цветовые модели. Наиболее известная среди них — *RGB* (красный, зеленый, голубой): любой цвет представляется состоящим из трех компонентов различной интенсивности красного, зеленого и синего (от 0 до 255, 255, 255, 255 — белый, 0, 0, 0 — черный). Каждый из основных цветов тем ярче, чем больше его частей входит в смесь.

Модель *CMYK* (голубой, фиолетовый, желтый, полутон или черный) также строится на смешении основных цветов в процентном соотношении к максимальному значению (белый цвет).

Модель *HLS* (оттенок, яркость, насыщенность) тоже использует деление цвета на составляющие, но не по основным цветам.

Цвета модели *PANTONE®* выбираются из соответствующего списка стандартных образцов или заданием соответствующего номера образца.

▲Графические форматы

В курсе «Работа с графическими объектами» учащиеся Лицея выполняют лабораторные работы по сканированию изображения, дальнейшей его обработке с целью получения либо изображения (как «слайда») для монитора высокого разрешения, либо «слайда» для монитора низкого разрешения, либо твердой копии изображения, полученной на лазерном или матричном принтере. Необходимость получения качественного изображения для определенных целей заставляет учащихся уже на этапе сканирования образа продумать наиболее эффективный в конкретной работе графический формат файла, оптимально возможный его размер, цветовое решение готового изображения.

Всегда важно найти оптимальный графический формат для дальнейшей обработки полученных графических объектов. При работе с графическими объ-

ектами в различных пакетах обработки изображений возникают проблемы переносимости изображений из одного пакета в другой и сжатия их с целью экономии дискового пространства.

Наиболее популярные графические форматы файлов — *PCX, GIF, TIFF, BMP, TARGA, JPEG.*

Формат PCX является «родным» для известного графического редактора ZSoft Paintbrush. PCX использует простейший способ сжатия изображений, позволяет выполнить быстрое копирование изображения из файла в память и обратно, хранит палитру в файле в представлении RGB, данные в представлении 1, 4, 8 и 24 бита на пиксел.

Формат BMP позволяет хранить данные, представленные в виде 1, 4, 8 и 24 бита на пиксел. Это стандартный формат файлов программы Windows Paintbrush, его «понимают» практически все Windows-приложения. Этот формат создавался для программ, работающих с буфером промежуточного хранения.

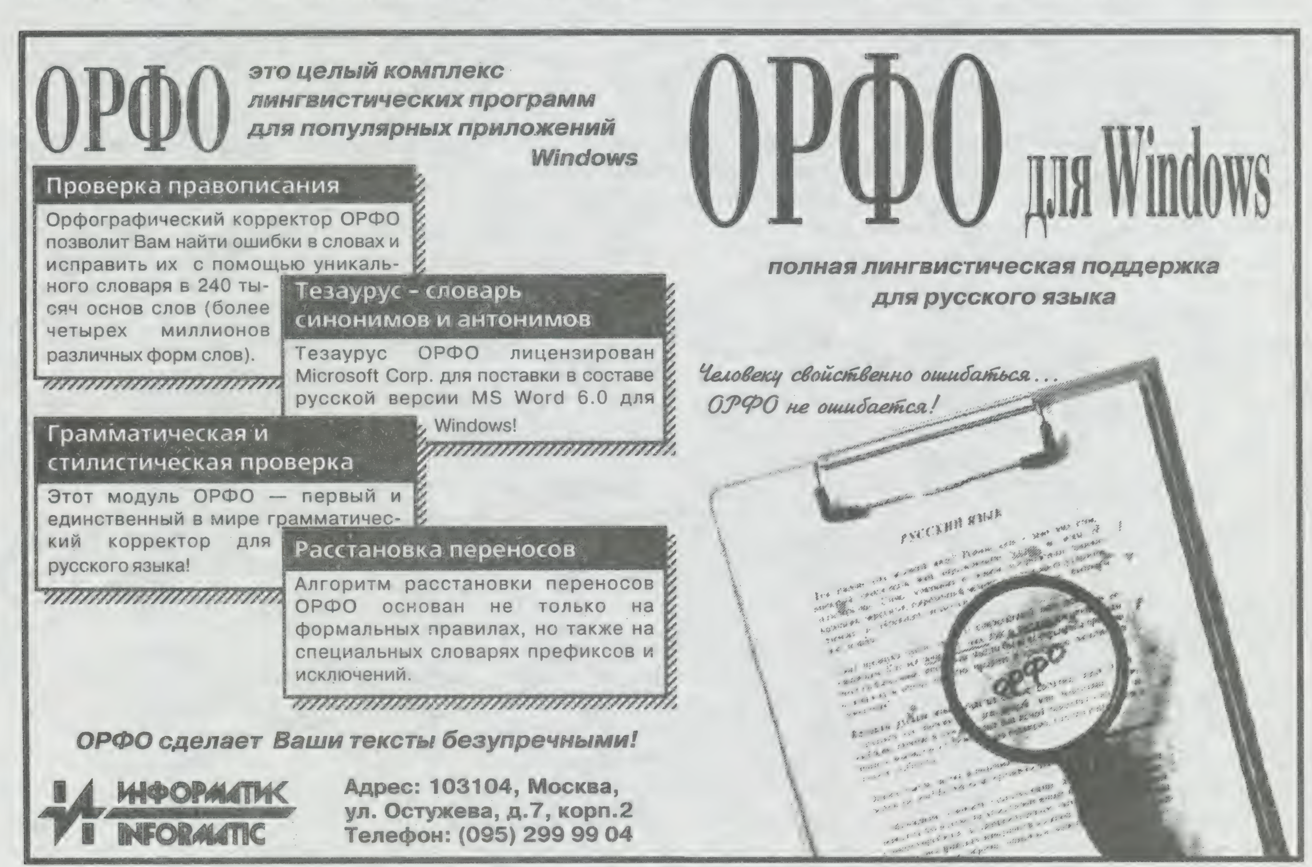
Формат GIF позволяет хранить данные, представленные в виде 1, 4 и 8 бита на пиксел, применяется в пакете Autodesk Animator. При достаточно простой структуре файла и наличии небольшого числа атрибутов изображения он использует более эффективный алгоритм сжатия, чем PCX. Так же как PCX, этот формат позволяет хранить палитру в файле; отличием его является точное соответствие цвета палитры цвету на оригинале, благодаря чему можно более полно переносить рисунок на различные типы мониторов.

Файловый формат TIFF поддерживает различные типы изображений: черно-белые, полутоновые и цветные, может использовать четыре типа упаковки данных, поддерживает данные, представленные в виде 1, 4, 8 и 24 бита на пиксел. TIFF имеет множество атрибутов, служащих для точного описания сложных изображений. Он также позволяет хранить данные о палитре в файле. Чрезвычайно популярный формат для издательской деятельности, TIFF «понимают» практически все издательские системы и графические пакеты.

Графический формат TARGA поддерживает данные, представленные в виде 1, 4, 8, 24 и 32 бита на пиксел. Он поддерживает различные типы изображений: полутоновые и цветные изображения. Может использовать три типа паковки. Этот графический формат дает изображения, наиболее приближенные к телевизионным.

Формат JPEG поддерживает данные, представленные группами по 8 и 24 бита на пиксел. Он является стандартом Объединения Профессиональных Фотографов. JPEG имеет эффективный алгоритм сжатия, но большое сжатие JPEG-изображения сопровождается существенной потерей качества.

(Продолжение следует)



SoundBlaster AWE-32 и студия внутри системного блока

Андрей Родионов

Уважаемый Андрей!

В одном из ваших ответов были такие строки: "А вообще-то лучше всего ставить карту AWE-32. Это идеальный вариант для любых приложений, включая профессиональную музыкальную работу".

Хотелось бы подробнее узнать про эту карту, ее возможности. Я сейчас ищу карту, которая позволила бы мне проигрывать готовые сэмплы как инструменты (MIDI или еще как-нибудь), но этот вопрос, как правило (к большому сожалению), опускается при обзоре той или иной карты. Где-то я читал про WaveBlaster, который якобы подсоединяется к SoundBlaster и позволяет что-то подобное делать, но как-то это все коряво выглядит, похоже, что буржуи просто хотят побольше выкачать денег. Хотелось бы узнать, существуют ли карты с требуемыми возможностями или нет? Я занимаюсь музыкой профессионально и ищу достойный внимания вариант. Буду рад любому комментарию.

С уважением, Сергей Б.

AWE-32 — это новая разработка Creative Labs. Она принципиально отличается от всех предыдущих карт семейства SoundBlaster тем, что может быть использована для профессиональных музыкальных приложений. У нее на борту находится ASP (Advanced Signal Processor), чип E-8000 от EMU Systems (тот же, что в Протеусах и проч. — 16 бит, 32 голоса полифония), 4 Мбайт ROM с готовыми сэмплерными основами и параметрами волнового синтеза для пресетных GM/GS/MT32 наборов, 512 Кбайт RAM (расширяемые двумя SIMM до фактических 28 Мбайт: 32RAM — 4ROM) для хранения загружаемых сэмплов и параметров волнового синтеза, 8-битный 20-голосный 4-операторный FM-синтезатор (AdLib-compatible, может быть "заведен" в хорус/ревер ASP), 16-битные стерео АЦП/ЦАП с частотой дискретизации до 48 кГц, линейный стереовход и стереовыход, микрофонный моновход с фантомным питанием, 4 Вт выход для пассивных динамиков, MIDI IN/OUT (как обычно, совмещенные с разъемом Joy-stick — требуется дополнительный шнур-адаптер с оптронной развязкой). На карте имеется разъем для дополнительного подключения WaveBlaster ("садится" параллельно AWE-32 MIDI OUT), четырех типов приводов CD-ROM, SPDIF-выход ASP и дополнительный вход для подключения двух проводов от PC-спикера. Карта снабжается всеми драйверами, необходимыми для работы под Windows, превосходными аппликациями на 8 дисках (последняя версия на 9 — добавлен долгожданный редактор волнового синтеза "Vienna"). И наконец, AWE действительно на все 256,5% совместима со всеми предыдущими моделями SoundBlaster (это означает, что все аппликации и игры ее "знают" и могут с ней работать). AWE-32, претендуя на то, чтобы стать промышленным стандартом, видимо, в самом деле скоро им станет, так как люди из Creative ОЧЕНЬ ПОСТАРАЛИСЬ и к тому же ведут замечательную политику цен/сопровождения.

Теперь немного о том, что можно делать с помощью AWE.

В моем рабочем компьютере (DX2-66/16 Мбайт) из звуковых карт в настоящий момент установлены Roland SCC-1 и Creative AWE-32 (с CD 563). Таким образом, общая полифония достигает 24+32+20 — всего 76 голосов, плюс еще 2 midi ins, 2 midi outs. В качестве MIDI-контроллера я использую замечательную 6-октавную клавиатуру Roland A-30 с их же (Роланд) педалью рояльного типа. WaveBlaster я выкинул, так как он, хотя и звучит неплохо (тоже чип от EMU, 4 Мбайт ROM и 32 голоса полифония), но не имеет своих собственных хоруса/ревера и, конечно, не "дотягивает" до SCC по динамике и выразительности звука. (Вообще, WaveBlaster был разработан в качестве GM-дополнения для обычных SoundBlaster, чтобы они звучали получше, а не только FM-ом.)

ОЧЕНЬ ВАЖНО то, что обе карты имеют отбор в свои собственные хорусы/ревербераторы из каждого MIDI-канала независимо, что позволяет иметь на выходе полноценную и полностью сведенную фонограмму, управляемую прямо из секвенсора (я, конечно же, использую Cubase от Steinberg). В качестве редактора для SCC-1 я использую мой собственный редактор, построенный средствами объектного конструктора Cubase, хотя, по правде говоря, все 354 тембра SCC настолько хороши и выразительны, что их и редактировать нет особой нужды. А для собственной звуковой палитры с недавних пор я пользуюсь очень удобным редактором Vien-





Андрей Родионов никак не укладывается в рамки традиционного в нашей рубрике краткого представления. Не знаю, как его назвать — то ли Multimedia Man то ли Человек-оркестр. Судите сами.

Опытный системно-прикладной программист на ЕС ЭВМ (ветераны ЕС помнят целую надстройку усовершенствований над ДОС ЕС производства едва ли не первой негосударственной софтверной "компании" Rodionov Software — состоявшей ровно из одного человека), потом на HP3000 (руководил разработкой программного комплекса большой библиографической системы для ИНИОН).

Композитор и исполнитель, автор музыки и песен ко многим спектаклям и мультфильмам, автор музыки и дизайна первых в российской практике грампластинок, полностью подготовленных на персональных компьютерах: "Пульс 1", "512 Кбайт" и "Персональный компьютер".

Автор и разработчик компьютерных игр для HP3000 и разных персональных компьютеров, начиная с MSX, а также графических, анимационных и музыкальных библиотек, системных расширений, инструментальных пакетов и авторских средств для разработки игр. Есть у Андрея сквозная любимая тема, фрагментами реализованная на разных платформах — мирная космическая одиссея с наделенными собственным поведением персонажами — бортинженером Фгрм-пыком, программистом Таней и другими. Автор "классики жанра" — статьи "Своя игра" (МирПК, №1'93) — лучшего, пожалуй, что я читал о компьютерных играх.

Президент АО Ракурс. Помимо общего руководства компанией, возглавляет отделение компьютерных технологий, отделы компьютерных игр и новых музыкальных технологий (заказные музыкальные рабочие станции).

Член правления Российского отделения Ассоциации электроакустической музыки при ЮНЕСКО, заведующий лабораторией электронной и компьютерной музыки при Союзе композиторов России.

При всем этом Андрей еще замечательный электронщик, схемотехник и просто техник-изобретатель. Представляете, какой коллектив в одном лице? Так что когда один из членов этой "команды" берется что-то делать, у него всегда есть надежная поддержка из смежных и не очень областей. И еще одно определяющее свойство Андрея, главный, по-моему, "двигатель прогресса" — это его стремление и умение со вкусом обустраиваться в любой новой среде, окружении. Все, будь то рабочий кабинет, операционная система, музыкальная студия или автомобиль, несет на себе неизгладимый отпечаток творческой личности хозяина и имеет на расстоянии вытянутой руки весь мыслимый и немыслимый сервис, как правило, придуманный и реализованный им же. То он объединяет все устройства и компьютеры студии в совершенно оригинальную локальную сеть на базе MIDI-протокола, позволяющую компьютерам разных платформ разделять ресурсы; то вдруг, устав дозваниваться по межгороду, берет компилятор BDS C и паяльник и превращает старенький 8-разрядный MSX в автоответчик-секретарь-планировщик. Можно вспомнить и целые списки замечательных утилит для многих ОС, в которых ему пришлось работать, и библиотеку собственных звуков с удивительными старинного звучания скрипками, и... всего не перечислишь, надо целую статью писать. Мне кажется, Андрей всю жизнь просто играет в Свою Игру. Когда много лет назад я пришел на диплом в ИКИ, Андрей уже работал там ночным оператором и системщиком (будучи днем студентом МЭИС) на новенькой ЕС-1020, заводской №17. Уже тогда, за десять лет до Макинтоша, он пытался соорудить из этого размером в зал прообраза персоналки (ОЗУ 64 Кбайт, диски по 7 Мбайт) интеллектуальную мультимедиа-приставку с дружественным интерфейсом: машина отбивала на всех внешних устройствах какую-то сложную синкопическую сюиту для ударных, писала стихи, вела с вами беседы на отвлеченные темы на своем дребезжащем Consul'e.

Поскольку в Андрее по очереди берут верх то композитор, то программист, то еще кто-нибудь, на неделю-другую почти полностью выключая его из остальных дел, мне никак не удается подвигнуть его на глобальный обзор сегодняшнего состояния дел в компьютерной музыке — в стиле "Своей игры". Поэтому в конце концов мы пришли к компромиссу и решили начать публиковать серию ответов Андрея на наиболее типичные вопросы, заданные ему в Релкоме, в конференциях по цифровому звуку и музыке. А там посмотрим.

па, который позволяет "раскручивать" как бортовой ROM, так и мои собственные сэмплы на мультилэйерной основе (хоть 32 слоя разнесенные/совмещенные любым образом по клавиатуре!). Каждый слой управляется двумя шестистадийными EG (с дилэйем в начале!) amp/filter/pitch, двумя LFO (pitch, pitch/waw/tremolo), имеет свои собственные ревер/хорус (в каждом слое свои!!! — и это помимо общего ASP!!!!!!!!!). То есть традиционная развитая Муговская схема с генераторами, фильтрами и амплифайрами в цифровом виде и с сэмплерной основой. EMU, конечно, пару "мелочей" забыли в

своем E-8000, но и того, что есть для большинства приложений синтеза, более чем достаточно. Во всяком случае более удобного в управлении (не только с экрана) синтезатора за многие годы работы с ними мне еще не попадалось. (Исключая, пожалуй, старый добрый Roland Jupiter-8, но в те времена еще не было даже MIDI, не говоря уже о сэмплерной основе звука.)

Учитывая то, что линейный выход SCC "заведен" у меня на линейный вход AWE (и тем самым на его программный экранный микшер), я получаю полностью сведенную инструментальную фонограмму на линейном выходе AWE. А запуская вместе с Cubase внешне простую, но очень удобную программу Creative DAT "WAVE", я могу делать хард-диск-рекординг сведенного инструментального материала. Результатом является (стерео/16 бит/44 кГц) Wav-файл (на специальном линейном участке диска БЕЗ Double Space или какой-либо иной компрессии!). Для редакции я использую 2-канальный WAVE от Turtle Beach или SAW & SAW Utilities Rack от Innovative Quality Software. Последний является 8-канальным цифровым магнитофоном, на который в оставшиеся 6 каналов я могу дописывать вокал и/или солистов-инструменталистов и редактировать материал с точностью до одного отсчета (сэмпла) прямо с HD. Естественно, что для последующего сведения 8-канальной фонограммы используются СОФТВЕРНЫЕ процессоры SAW: эквалайзер, компрессор/лимитер, задержки, фэйды и пр. (Существуют и другие технологии, например, когда секвенсор все время "играет" синхронно с хард-диск-рекордером, который пишет/воспроизводит солистов на все 8 каналов, и это в принципе удобней, так как окончательное сведение полнее редактируется — но требуется гораздо больше ресурсов по быстродействию, а еще лучше — второй компьютер.) Все это имеет связь с внешним миром и работает через 16-битные стерео АЦП/ЦАП AWE и в случае необходимости может быть дополнительно откорректировано его собственным экранным микшером, использующим on-board ASP.

Интересно то, что для своей собственной звуковой палитры я могу использовать не только ROM AWE или собственные звуки, сэмплированные через AWE, но и любую музыку с CD-drive, переписанную хард-диск-рекордером на HD. Кто "у нас" много и хорошо работает со звуком? Jarre, например? Пожалуйста... "Вырезаем" на экране любые куски фонограммы, редактируем/обрабатываем софтверными процессорами там же, сохраняем в виде моно-WAV-файлов (16 бит, 44 кГц) и загружаем их в качестве сэмплерной основы в user-bank AWE вместе с другими RAM/ROM-сэмплами, редактируем средствами Vienna... Используем в Cubase как один из user-банков... Комментарии, как говорится, излишни.

Обратите внимание, что сегодня практически ВСЯ студия может быть ВНУТРИ системного блока — и это очень удобно. По правде говоря, сам я никогда в своей практике не работал столь комфортно, как после погружения всей студии в компьютер. Правда, компьютер требуется мощный, надежный и с большими ресурсами как по RAM, так и по HD. Для уверенной работы одновременно секвенсора и хард-диск-рекордера нужны очень быстрый большой диск и высокопроизводительный центральный процессор. Здесь иногда может не хватать и DX2-66. Сам я подумываю сейчас о DX4 (not Pentium with its FDIV bug!) Кстати, никакое программное/аппаратное кэширование дисков для хард-диск-рекординга не помогает, так же, впрочем, как и SCSI (это отдельный разговор). Требуется ДЕЙСТВИТЕЛЬНО быстрый IDE с 5-8 Мбайт/с (для многоканальной записи). За пределами нашего рассмотрения остались как все внешние звуковые модули (т.н. "мозги"), так и множество других звуковых карт — всевозможные SoundBlaster и их клоны, целые семейства Turtle Beach, Ultrasound, другие карты Roland, EMU, Yamaha и пр. Но не потому, что они нам неизвестны — практически все карты тестировались (на декабрь 1994 года), но не удовлетворяли тем или иным требованиям (техническим, музыкальным или "разумности" по цене), и это тоже отдельный большой разговор, которому не место в данном кратком комментарии. Единственной достойной, на мой взгляд, новинкой является новая разработка Роланд — та же самая SCC, но выполненная в конструктиве WaveBlaster, то есть "надеваемая" на разъем карт семейства SoundBlaster 16 (в том числе AWE), а не вставляемая в отдельный слот.

В конце своего комментария, как это ни печально, должен предупредить, что если вы решитесь построить сами аппаратно-программную конфигурацию, подобную вышеописанной, вас могут не удовлетворить результаты самостоятельной увязки/настройки всей аппаратуры/софтвера, даже если вы — опытный пользователь или программист. Дело в том, что системная интеграция сложной многокомпонентной системы — дело очень тонкое, требующее большого опыта и знаний как в области компьютеров и системного софта вообще, так и конкретной аппаратуры/аппликаций в частности. Особенно когда аппаратура работает в режиме реального времени, близком к пределу производительности системы (сложные партитуры в секвенсоре, хард-диск-рекординг, синхронизация аппликаций и пр.). Тем не менее желаю вам успеха.

e-mail: root@rodionov.msk.ru

НОВЕЙШАЯ КОМПЬЮТЕРНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ



СИСТЕМНАЯ ИНТЕГРАЦИЯ

- Комплексные решения в области создания корпоративных информационных систем
- Интеграция имеющихся информационных ресурсов в единую систему
- Разработка и моделирование сетевых проектов в нашей лаборатории
- Оперативная техническая и информационная поддержка
- Надежное гарантийное и послегарантийное обслуживание
- Обучение персонала и консультирование
- Предварительное апробирование решений

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ОТКАЗОУСТОЙЧИВЫЕ СЕРВЕРЫ

- Серверы семейства ProLiant фирмы COMPAQ
- Модульные процессорные карты 486/Pentium с архитектурой SMP позволяют создавать наращиваемые 2- и 4- процессорные комплексы с симметричной обработкой
- Оперативная память до 512 МБ с корректирующими кодами
- Автономная система диагностики и контроля в процессе работы со встроенным SNMP-агентом
- Compaq TriFlex с пропускной способностью 267 МБ/с.

ЛОКАЛЬНЫЕ И КОРПОРАТИВНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ

- Оборудование Ethernet 10/100, Token Ring, FDDI, ATM
- Многофункциональные модульные концентраторы
- Мосты и мультипротокольные маршрутизаторы
- Телекоммуникационное оборудование
- Диагностическое оборудование: кабельные сканеры, рефлектометры, протокольные анализаторы
- Средства администрирования, мониторинга и управления на базе SNMP

МНОГОУРОВНЕВЫЕ СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

- Дисковые массивы RAID-5
- Магнитооптические накопители и роботизированные библиотеки
- Стримеры QIC, DAT, DLT, 8mm Helican Scan
- Ленточные библиотеки CD-ROM: Readers, CD-Recorders, Changers, Premastering Software
- Интегрированные системы архивации и миграции данных в среде NetWare, Unix, Windows NT

COMPAQ
Авторизованный системный партнер

Россия, 103050, Москва, ул. Ермоловой, 22, стр. 1
Тел.: (095) 200-1696, 299-4307 Факс: (095) 299-4625





ElectricImage Animation System 2.1

Константин Федотчев

Появление процессоров PowerPC и построенных на их основе компьютеров PowerMacintosh не позволяет станциям Silicon Graphics поставить точку в споре о том, какая платформа лучше при создании компьютерной графики. Сегодня маковские программы трехмерной графики и анимации, написанные для старой платформы, адаптированы, переписаны, оптимизированы под RISC-процессор. Причем большинство производителей программ не изменили цену. Изменилась производительность. Cosa After Effects стала в 6 раз быстрее, Form-Z в 15 раз, ElectricImage в 22 раза, Animation Stand в 36 раз. Таким образом, графические станции на базе новой серии Macintosh по производительности сравнялись со станциями на основе недорогих моделей SGI. При этом, если сравнить суммарные стоимости оборудования и программного обеспечения этих станций, то окажется, что при одинаковой производительности цена Маковской станции раза в три меньше Силиконовой — поскольку разрыв в стоимости программных средств для Мака и Силикона даже больше, чем в стоимости собственно компьютеров и дополнительных плат.



У сторонников SGI козырной картой в игре за превосходство всегда были возможности анимации. При этом, когда сравнивали станции для телевидения, всегда говорили: "А вот в "Терминаторе" использовали Силикон потому, что только у него такие возможности и только он так быстро считает". Если взглянуть на это утверждение более детально, то окажется, что "суп не отделен от мух". Во-первых, между кино и телевидением огромная разница. Для телевидения разрешение в 1000 элементов в строке уже избыточно. В кино 5000 элементов в кадре по горизонтали минимально. Поэтому при создании графики с киноразрешением всегда используют несколько станций. Например, в клипе Питера Габриеля Steam для создания эпизода с огненными танцорами длительностью менее десяти секунд пришлось использовать семь компьютеров серии Iris. Во-вторых, большинство эпизодов в кино создаются, как правило, не только и не столько на стандартных коммерческих программах, сколько на "штучном" софте, являющемся собственностью студий, таких как ILM (и который, естественно, поставщики станций SGI предоставить не могут). Но в нашей стране основным заказчиком компьютерной графики является телевидение. Это рекламные ролики и оформление эфира (шапки, за-

БЫСТРО. ПРОСТО. НАДЕЖНО.

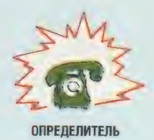


















Модем / Факс / Автоответчик / Определитель номера

Все модемы ZyXEL совместимы с большинством других модемов и могут работать по коммутируемым или выделенным линиям в синхронном или асинхронном режиме с такими программными средами, как: DOS™, Windows®, OS/2®, Macintosh®, UNIX®, NeXT®, Amiga™, Atari™. В синхронном режиме с использованием команд V.25bis ZyXEL совместим с системами AS/400® и RS/6000® фирмы IBM.

- **Высокая скорость – 19200 бит/с**
- **Сверхнадежный режим ZyCELL**
- **V.32bis/V.32, V22bis/V.22, BELL 212A**
- **V.17 14400 бит/с, CLASS 1, CLASS 2/2.0, G3 Факс**
- **V.42/V42.bis (+ Selective Reject), MNP® 3/4/5**
- **Цифровая запись/воспроизведение звука**
- **Распознавание условных звонков**
- **Определение номера вызывающего абонента**
- **Улучшенное распознавание сигналов АТС**
- **Дистанционное конфигурирование**
- **Динамический выбор рабочей скорости**
- **Защита от НСД: пароли, обратный звонок**
- **Адаптация к абонентской линии**
- **Регулировка уровня передачи на коммутируемой линии до 0 дб**
- **2/4 проводная коммутируемая/выделенная линия**
- **Автоматическое распознавание вызова модем/факс/голос**
- **Перепрограммируемое ПЗУ – обновление микропрограммы**
- **Документация и программа на русском языке**

Приобретая модемы ZyXEL у авторизованных дилеров МКЦ "Вариант", Вы получите оборудование, произведенное специально для России, имеющее сертификат Министерства связи, неоходимые дополнительные программы, 2 года гарантии со склада, бесплатное обновление микропрограммы, профессиональную поддержку квалифицированного технического персонала и доступ к BBS.



**МОДЕМЫ,
КОТОРЫЕ ОБЪЕДИНЯЮТ
РОССИЮ**

**Официальный дистрибьютор
АО МКЦ "Вариант"
117279 Москва, ул. Островитянова, 37а
Тел. (095) 420 2519
Факс (095) 420 5311**

**Информация (095) 932 8510
Техническая поддержка (095) 932 7201
(095) 932 7601
WHITE BEAR BBS (095) 932 8465
zyxel@variant.msk.su, 2:5020/22@fidonet**



ставки, перебивки). Для этих целей уникальные способности Силиконов чаще всего остаются невостребованными.

Давайте посмотрим на примере новой версии программы ElectricImage Animation System 2.1, какие возможности по трехмерной анимации может иметь станция на базе Power Macintosh. Вся программа состоит из нескольких модулей. Первый модуль позволяет сделать то, что очень любят заказчики рекламы в нашей стране, — название компании золотыми трехмерными буквами. Точнее, сделать трехмерную модель из обычного PostScript-шрифта. Причем при особом желании заказчика буквы будут с фасками. Меняя прозрачность или материал фасок и самих букв, можно в дальнейшем получить вспышку или блеск на этих фасках.

ElectricImage 2.1 может без предварительной конвертации читать около двадцати типов моделей. Иными словами, можно использовать модели от всех существующих программ для всех платформ, включая всенародно любимую 3D Studio. Так же просто осуществляется импорт.

После того как модель попадает в EI, имеется возможность связать объекты по любым из восьми приоритетов, что по возможностям соответствует скелетной анимации. Объекты можно также разными способами деформировать: сжимать, растягивать, скручивать, изгибать или использовать Безье-деформации.

В программе появились ранее закрытые для рядовых пользователей модули — например, позволяющие взрывать объекты на множество осколков, число которых задается художником. Можно регулировать параметры взрыва: сопротивление воздуха, завихрения, время вспышки. В другом модуле можно получить такие сложные эффекты, как летающие искры, пузыри и фонтаны.

При работе с текстурами можно регулировать величину отблеска, прозрачность, степень отражения и другие физические свойства. Параметры текстур могут быть оптимизированы при расчете. Так, границы объекта могут быть просчитаны с большей четкостью и сглаживанием. Можно получать муары, свойственные некоторым материалам.

Все стандартные источники света в сцене могут быть дополнены неоновым светом и лучом лазера. Благодаря удобному меню анимации и этим возможностям работы со светом в ElectricImage очень легко создавать вспышки и вообще любые моментальные события.

В программу могут быть подгружены звуковые файлы, которые отражаются в меню анимации в графическом виде. Звук может автоматически синхронизироваться с анимацией, которую можно посмотреть в реальном времени в QuickTime-формате.

Модуль рендера ElectricImage давно заслужил признание благодаря быстроте счета, которая достигается во многом за счет изощренной оптимизации. Приведем несколько цифр, показывающих, насколько серьезна эта программа:

- максимальное число сегментов 30 000 000;
- максимальное число групп моделей, текстур, источников света — по 4 000 000.

Благодаря вышеуказанным возможностям ElectricImage Animation System получила заслуженное признание. В кино эта программа использовалась в Terminator 2, Jurassic Park, The Mask. На телевидении EI применяется очень широко — от учебных фильмов для кабельных сетей до оформления последней зимней Олимпиады.



Наша Серия


оптимальный выбор
ДИЛАЙН


горячая линия
ДИЛАЙН


прямо со склада
ДИЛАЙН




ваш заказ
ДИЛАЙН


всегда в срок
ДИЛАЙН


поддержка дилеров
ДИЛАЙН

Там, где пехота не пройдет, или Несколько слов о самолетных имитаторах

Мл. сержант ВВС

Алексей Федоров

На этот раз мы рассмотрим несколько самолетных имитаторов, появившихся у нас в продаже. Но сначала я позволю себе небольшое введение.

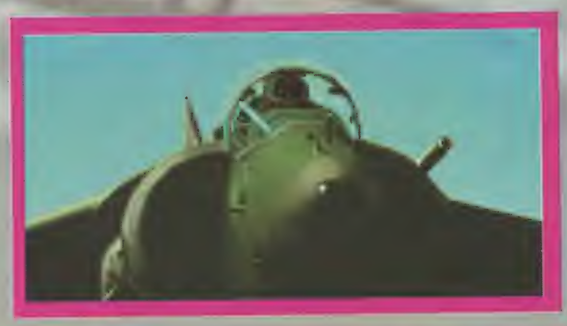
Надо отметить, что самолетные имитаторы — один из наиболее популярных жанров компьютерных игр. По числу выпускаемые имитаторы порой превосходят игры других чрезвычайно популярных жанров, таких как приключенческие и аркадные игры. При этом имитаторы самолетов времен Второй мировой войны занимают примерно 35%, имитаторы современных самолетов — 35%, и 30% приходится на самолеты Первой мировой войны и различные экспериментальные модели. Такой феномен довольно легко объяснить: в самолетных имитаторах играющий может делать то, что не всегда удается в реальной жизни: управлять самолетом, участвовать в боевых операциях, решать исход крупнейших сражений. Немаловажной причиной интереса к таким играм (производимым в основном ведущими игровыми фирмами США) является недавний ряд успешных операций ВВС США на различных театрах военных действий. Игры такого типа можно разделить на несколько видов, самый интересный из которых — имитаторы, максимально повторяющие процесс управления самолетом. Такие имитаторы могут использоваться даже для тренировки пилотов. К другому типу относятся игры, позволяющие летать на различных самолетах, управление которыми максимально упрощено — например, имеется одна кнопка для запуска двигателей, а сам полет управляется клавишами управления курсором.



Наиболее часто фирмы-производители выпускают имитаторы современных самолетов, но также встречаются модели самолетов времен Второй и даже Первой мировых войн. Среди последних можно выделить сериал **Great War Planes** фирмы Dynamix (партнера фирмы Sierra On-Line), который состоит из имитаторов самолетов тех лет: A-10 Tank Killer, Red Baron (включая средство для построения собственных миссий Mission Builder), Aces of the Pacific и Aces Over Europe. В скором времени планируется выпуск еще одного имитатора — Aces Over Korea.

Другой известной фирмой, выпустившей множество самолетных имитаторов, является **MicroProse**, продукты которой посвящены наиболее известным моделям самолетов Второй мировой войны, а также современным самолетам. Многие самолетные имитаторы имеют продолжения — дополнительные миссии, дополнительные самолеты, которые объединяются вместе с оригинальными играми и часто выпускаются на компакт-дисках (CD-ROM). У нас в продаже имеются такие имитаторы, как F-15 Strike Eagle III и Gunship 2000 фирмы MicroProse, Comanche : Maximum Overkill фирмы NovaLogic, Secret Weapons of the Luftwaffe фирмы LucasArts, Strike Commander фирмы Origin, и ряд других. Большинство из них поставляется на CD-ROM. Наиболее популярным имитатором гражданских самолетов можно смело считать Flight Simulator 5 фирмы Microsoft. Для него выпущены и имеются у нас в продаже дополнительные сценарии — полеты над Нью-Йорком (Microsoft New York) и Парижем (Microsoft Paris).

Те, кого не удовлетворяют готовые имитаторы, могут попробовать свои силы в изготовлении собственных. Необходимо только приобрести Flight Simulator Toolkit фирмы Domark и найти много свободного времени. В результате вы получите самолетный имитатор, созданный собственными руками. Flight Simulator Toolkit представляет







собой набор редакторов, позволяющих не только разработать собственную модель самолета (или, если вы знаете технические характеристики, то повторить какой-либо настоящий самолет), но и разработать приборную доску, создать ландшафт и проверить аэродинамические характеристики созданного самолета. Набор средств, входящих в Flight Simulator Toolkit, используется самой фирмой для разработки самолетных имитаторов.

Среди изготовителей самолетных имитаторов также известна фирма **Spectrum Holobyte**. Сборник лучших имитаторов этой фирмы Falcon 3.0 Gold появился у нас в продаже в конце осени прошлого года, и на его примере мы более подробно посмотрим, что представляет собой имитатор самолета. Сборник Falcon 3.0 Gold может представлять интерес не только для любителей, но и для профессионалов. В его состав включены одни из лучших имитаторов самолетов, когда-либо выпускавшихся для компьютеров. Базовым самолетом выбран F-16, стоящий на вооружении многих стран, а также МиГ-29 и F/A-18 Hornet. Дополнительно в этот сборник включен видеокурс Art of the Kill, который может служить начальным руководством при ознакомлении с основными принципами управления современными боевыми самолетами. Сама игра Falcon 3.0 состоит из нескольких разделов. В первом вы можете просто попробовать свои силы в управлении самолетом. Во втором разделе вам предлагается подробный курс, посвященный всем аспектам полетов на F-16, включая взлет, посадку, заход на цель, бомбометание, форсаж и т.п. Вы научитесь понимать показания приборов, расположенных в кабине, и освоите основные принципы действий в воздухе. После этого можно переходить к третьему разделу — имитации боевых кампаний — Кувейт, Израиль или Панама. Вам предлагается принять участие в настоящих боевых действиях и наконец-то столкнуться с противниками. Более чем 300-страничное пособие, поставляемое с этой "игрой", написанное профессиональными военными летчиками, содержит всю необходимую информацию — от основ аэродинамики до истории создания самолетов серии Falcon и подробного описания театров военных действий, включая крупномасштабные карты. Отдельный интерес для специалистов может представлять приложение, в котором приведены тактико-технические характеристики различных самолетов и вертолетов, а также вооружения. Имитатор отечественного самолета МиГ-29 является дополнительной к Falcon 3.0 игрой и позволяет посмотреть на развитие современной авиации с другой стороны — со стороны противника. Созданный в конце 70-х, МиГ-29 до сих пор относится к одной из самых мощных боевых машин, и интерес к этой модели далеко не случаен. Документация содержит подробное описание процесса управления этим самолетом, и единственным ее недостатком является то, что все характеристики приводятся не в метрической системе. Этот имитатор, как, впрочем, и 3.0, можно смело использовать на начальных курсах военных училищ для ознакомления с базовыми характеристиками самолета.

Завершая этот вводный обзор, необходимо сказать несколько слов об аппаратных требованиях к таким играм. Они довольно высоки, зато вы получаете игры, созданные с применением передовых технологий: 3-мерного моделирования, рэндеринга и искусственного интеллекта. Очевидно, что необходим компьютер не ниже 80386 с тактовой частотой 33-40 МГц, имеющий 4-8 Мбайт памяти, же-



сткий диск объемом не менее 200 Мбайт, а также желательно наличие 8- или 16-битной звуковой карты, совместимой с SoundBlaster, и привода CD-ROM. Если вы планируете всерьез заниматься самолетными имитаторами, то вам потребуется джойстик — специальное аналоговое устройство ввода, напоминающее рукоятку управления самолетом. Самые простые джойстики представляют собой вертикальную ручку, которая может перемещаться в различных направлениях, и несколько кнопок. Более специализированные джойстики, такие как Aviator фирмы QuickShot, Virtual Pilot фирмы CH Products или ThrustMaster фирмы ThrustMaster Inc., содержат набор дополнительных кнопок (например, "выбор оружия"), и большинство самолетных имитаторов ориентированы именно на эти модели. Помимо этого, существуют всевозможные ножные педали, шлемы и другие атрибуты, делающие компьютерные самолетные имитаторы максимально приближенными к реальным имитаторам.



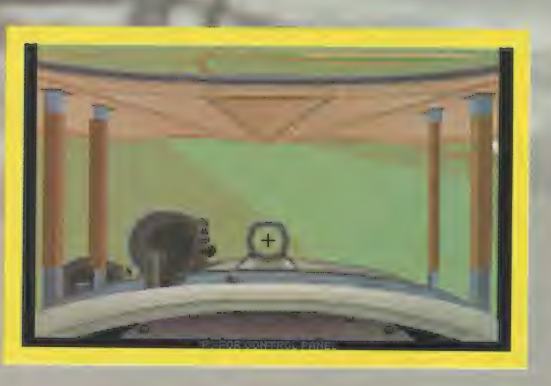
После того как мы познакомились с положением дел в области самолетных имитаторов в целом, давайте посмотрим на игры-имитаторы, отобранные для сегодняшнего обзора. Все рассматриваемые сегодня имитаторы — это продукция известных фирм Three-Sixty Pacific, Dynamix, Ocean и Domark. Так как самолетные имитаторы, представленные этими фирмами, покрывают практически весь этап развития авиации XX века (так уж получилось в нашем обзоре), начнем с имитатора Blue Max фирмы Three-Sixty Pacific — он посвящен самолетам Первой мировой войны.

Первая мировая война вошла в историю как война, в которой впервые применялась авиация. Так что у вас есть возможность перенестись к истокам развития военной авиации.

Blue Max, Three Sixty Pacific, 1991

На этот имитатор интересно взглянуть с нескольких сторон. Во-первых, в нем реализованы модели самолетов времен Первой мировой войны ("этажерки"), что не так уж часто встречается среди самолетных имитаторов. Во-вторых, этот имитатор не требователен к компьютеру и может работать в 1 Мбайте памяти (требуется всего 512 Кбайт) на компьютере с процессором 80286. Вы можете выбрать четыре модели самолетов союзников (если вы выступаете на их стороне) — RAF Se5a, Sopwith Camel, Nieuport Nie 17 или The Spad VII или четыре модели германских самолетов — Albatros D III и Fokker DV II, DR I и EIII. Возможно несколько вариантов полетов — практика, лобовая атака и кампании. Также можно играть вдвоем, что существенно интереснее.

Самолеты Первой мировой войны не были оборудованы радарами и всякими другими электронными приспособлениями, да и были неповоротливы, поэтому, чтобы сбить самолет противника, вам придется повозиться — несколько раз заходить на цель, приближаться практически вплотную и выделывать различные фигуры. С точки зрения графики вы не найдете в этой игре ничего необычного — детально прорисованная панель управления и векторный ландшафт, но возможность испытать самолеты, о которых можно узнать только в музеях или из учебников истории, должна привлечь ваше внимание.



Требования к компьютеру: 80286, 1 Мбайт памяти, VGA, привод CD-ROM, звуковая карта, совместимая с SoundBlaster, управление с клавиатуры, мышью или при помощи джойстика.

Dawn Patrol, Rowan Software, 1994

Эта игра не совсем обычна. Ее скорее можно назвать интерактивной энциклопедией. Эта энциклопедия и организована как книга — вы можете листать ее страница за страницей и изучать историю Первой мировой войны, биографии известных пилотов, модели самолетов и самые известные битвы. Практически на каждой странице есть иллюстративная вставка — графическая иллюстрация или анимация, а также специальная кнопка, которая переносит вас в режим имитации. Прежде чем начать полет, вы выбираете, на чьей стороне будете выступать, уточняете параметры боевого задания и сами задаете число врагов. Боевая миссия может быть прервана в любой момент и вы можете продолжить знакомство с энциклопедией. А если

воспользоваться встроенным магнитофоном для записи своих действий, то можно продолжить выполнение миссии с того момента, на котором она была прервана.

Игра дает вам возможность испробовать себя в роли одного из известных асов (по выбору) и принять участие в нескольких реальных боевых вылетах вместо него. Не менее увлекательно попробовать миссии, специально рассчитанные на ту или иную модель самолета, — а всего в энциклопедии их 14. Освоив эту модель в действии, вы можете выбирать ее в других миссиях.

В энциклопедию Dawn Patrol внесены сведения о пилотах с вымышленным именами. Одним из них можете стать вы. В ходе игры вы будете пополнять собственную биографию — вносить туда записи о победах и поражениях.

В режиме энциклопедии игра Dawn Patrol управляется манипулятором "мышь". С его помощью вы листаете страницы, выбираете миссии и устанавливаете необходимые параметры. В режиме самолетного имитатора управление самолетом производится клавишами управления курсором, функциональными клавишами и рядом других клавиш, описанию назначения которых в документации отведено несколько страниц.

На разработку игры Dawn Patrol было затрачено более пяти человеко-лет. Эта игра уникальна хотя бы тем, что в ней содержится огромный информационный материал и используется самая современная технология создания самолетных имитаторов.

Для этой игры требуется: компьютер с процессором 80386 с тактовой частотой не ниже 33 МГц, 2 Мбайт памяти, графический адаптер VGA, звуковая карта, совместимая с SoundBlaster, и привод CD-ROM.

Теперь мысленно перенесемся в Европу, где бушует Вторая мировая война. И посмотрим на имитатор Aces Over Europe фирмы Dynamix, входящий в серию имитаторов Great War Planes.

Исход Второй мировой войны во многом решился благодаря совместным усилиям армий союзников на европейском театре военных действий. И не последнюю роль в этом сыграла авиация.

Aces Over Europe, Dynamix, 1994

Имитатор "Асы над Европой" посвящен событиям завершающего периода Второй мировой войны. Игра уникальна тем, что дает возможность принять участие в имитации реальных сражений, которые происходили с апреля 1944 по май 1945 года. Она позволяет испробовать в действии более 20 моделей самолетов этого периода, стоявших на вооружении германского Люфтваффе, британских Королевских Воздушных Сил или американских ВВС.

Вы сможете вступить в воз-



душные дуэли с легендарными асами. Но перед такой исторически ответственной схваткой лучше провести тренировочные вылеты. В этой игре вы можете испытать свои силы в разнообразных миссиях двенадцати типов. Можно на выбор участвовать в каждой из них отдельно, а можно начать карьеру боевого пилота, — продвигаться от миссии к миссии, завоевывая звания и награды.

Игра может управляться с клавиатуры или с помощью джойстика. Основные клавиши — клавиши управления курсором, клавиши "<" и ">", а также клавиши от 1 до 9, управляющие мощностью двигателя, клавиша "А" — автопилот, "пробел" — стрельба из оружия, которое выбирается клавишей "G".

С помощью клавиши "Ввод" осуществляется переключение точек зрения, а их изменение — с помощью клавиш от F1 до F6. Разработчики предусмотрели даже специальную клавишу, помогающую в тех случаях, когда заклинивает пушку или пулемет. Это — клавиша "U".

В этой игре вы сможете управлять не только самим самолетом, но и условиями, в которых действует ваша боевая машина. Это — погода, степень видимости, количество патронов и горючего, отключение инструкций по ходу игры и так далее. Создание экстремальных ситуаций и выход из них — все в ваших руках.



При этом миссии могут быть записаны на встроенный "магнитофон" и затем проиграны для анализа. Вернуться в режим полета и переиграть миссию можно в любой момент.

Надо сказать, что разработчики из фирмы Dynamix очень тщательно подходят к созданию имитаторов. Воссоздание самолетов 50-летней давности — это довольно кропотливое занятие и требует особого подхода. Изучались документы, воспоминания пилотов, архивные материалы и уцелевшие копии реальных самолетов. Для этого имитатора был выбран не совсем стандартный графический режим, дающий в два раза больше пикселов по вертикали. В результате игра "Асы над Европой" просто поражает своей реалистичностью.

Требования к компьютеру: 80486SX, 4 Мбайт памяти, VGA/VESA, привод CD-ROM, звуковая карта, совместимая с SoundBlaster, рекомендуется использование джойстика.

Наша справка: фирма Dynamix является бизнес-партнером фирмы Sierra On-Line и известна самолетными имитаторами Red Baron, Aces of the Pacific, имитатором подводной лодки Aces of the Deep, ролевой игрой Betrayal at Krondor и рядом других.

После того как мы побывали в небе Европы во время Второй мировой войны, перенесемся в сегодняшние дни. Следующие два имитатора разработаны фирмой Domark Group Ltd., а выпущены фирмой Accolade в сборнике CD Compendium.

Super-VGA Harrier, Domark Group Ltd., 1993

Имитатор SVGA Harrier — это ремейк имитатора AV-8B Harrier фирмы Domark, работающий в режиме SVGA 640x480, 256 цветов. Документация, поставляемая с этой игрой, как, впрочем, и с MiG-29M (оба этих имитатора поставляются на одном диске), не содержит привычных для самолетных имитаторов иллюстраций, например изображения кабины и основных маневров, которые можно выполнять на самолете, а также каталога оружия и другой технической информации. Честно говоря, без каталога оружия или технических характеристик самолета — размаха крыльев, взлетного веса и т.п. можно прожить, но вот отсутствие изображения приборов в кабине затрудняет освоение этих имитаторов.




ELCO
TECHNOLOGY
ДИСТРИБЬЮТОР
NOVELL, APC
ОФИЦИАЛЬНЫЙ ПАРТНЕР
COMPAQ, CABLETRON,
MICRODYNE
ВИРТУОЗЫ
КОМПЬЮТЕРНЫХ СЕТЕЙ
NetWare
NOVELL NetWare
Microdyne
Москва. Центральный офис. Тел.: (095) 131-5555, 131-8383. Факс: (095) 131-1684.
Отделение "ЭЛКО-Университет". Тел.: (095) 939-4684. Факс: (095) 932-8821.
Санкт-Петербургское отделение. Тел.: (812) 277-7175. Факс: (812) 277-5807.
Ярославское отделение. Тел.: (0852) 27-5949



MiG-29M Superfulcrum, Domark Group Ltd., 1991

В этой игре вам отводится роль пилота одного из лучших в мире боевых самолетов — отечественного МиГ-29. Причем здесь вы выступаете в роли союзника американцев. Вспомним игру MiG-29: Deadly Adversary of Falcon 3.0 фирмы Spectrum Holobyte, где вы уничтожали американцев. Времена меняются, оттепель, разрядка...

А теперь посмотрим на имитаторы еще трех самолетов, два из которых находятся пока в стадии разработки и поступят на вооружение только в конце 90-х годов. Да, бывает и так!

TFX, Digital Image Design, 1993

В эту игру, полное название которой Tactical Fight Experiment, включены имитаторы трех самолетов, из которых только F-117A стоит на вооружении ВВС, а два других — EuroFighter 2000 и F-22 SuperStar появятся в воздухе только в конце 90-х годов. Для всех трех моделей самолетов используется общая приборная доска, отвечающая последним требованиям эргономики (если этот термин применим в военной промышленности). Управлять самолетами (самолетом, так как у них единая приборная доска) достаточно легко, в основном за счет того, что приборная доска не перегружена информацией. В отличие от многих других самолетных имитаторов здесь задействован только ряд клавиш клавиатуры — это функциональные и буквенные. Так что использование специальной накладки не требуется. Особенно, если этот имитатор не первый в вашей жизни и вы имеете приблизительное представление о том, как управлять самолетом. В игре используется векторная графика без каких-либо особых инноваций, но, может быть, это и к лучшему — игра довольно быстро работает даже на 33-мегагерцевом 386-м компьютере. Из нечасто встречающихся опций интересна возможность дозаправки самолетов. Но это так, "прибамбасы". В игре предлагается семь типов миссий — перехват, уничтожение противника в воздухе, уничтожение наземных целей, поддержка наземных операций с воздуха, уничтожение военных кораблей противника. Театры военных действий: Сомали, Колумбия (борьба с наркобаронами — сюжет, известный по имитатору ATAC фирмы MicroProse), Ливия, Югославия и Южная Атлантика.



Для желающих расширить свой кругозор. Самолет EuroFighter 2000 разрабатывается в Германии в рамках европейской программы по созданию мощного военного самолета. Он предназначен в первую очередь для воздушного боя. Его скорость — 2 Маха (то есть две скорости звука). Эта модель обладает автоматическим регулятором угла атаки и рядом других электронных приборов, делающих полет более безопасным. Модель F-117A — это американская разработка (компания Локхид). Самолет способен развивать скорость до 0,85 М и предназначен для воздушных атак. Отличительной особенностью этой модели является практически полная невидимость для радаров дальнего и ближнего обнаружения. F-22 SuperStar — еще один самолет фирмы Локхид. Он может развивать скорость до 2,3 М и, так же как и другие модели, включенные в комплект TFX, в первую очередь предназначен для воздушного боя.

Требования к компьютеру: процессор 386 и выше, 4 Мбайт памяти, звуковая карта, совместимая с SoundBlaster, привод CD-ROM, рекомендуется использование джойстика.

Игры для данного обзора предоставлены АО "ЮниВер", тел./факс (095) 434-20-60

DELL™ ПРОСТРАНСТВО УСПЕШНОГО БИЗНЕСА

OMNIPLEX
система
для масштабных
приложений

POWEREDGE SP/XE
мультипроцессорные
суперсерверы
для корпоративных
сетей

OPTIPLEX 560/XL
универсальная
офисная станция

OPTIPLEX X590
Мощная
графическая
станция

LATITUDE XP
беспрецедентное
время автономной
работы

OPTIPLEX 486/LE
недорогая
офисная станция

intel inside pentium PROCESSOR

IBS РЕКЛАМА

DELL COMPUTER Corp. ПРЕДЛАГАЕТ **ВСЕ ТИПЫ КОМПЬЮТЕРОВ** — ОТ ЛЕГКОЙ НЕДОРОГОЙ СТАНЦИИ ДО МНОГОПРОЦЕССОРНОГО СУПЕРСЕРВЕРА. ПРИОБРЕТАЯ КОМПЬЮТЕР, СОБРАННЫЙ **ТОЧНО ПО ЗАКАЗУ**, ПОЛЬЗОВАТЕЛЬ DELL ПОЛУЧАЕТ **ШИРОЧАЙШИЙ ВЫБОР** ИЗ БОЛЕЕ ЧЕМ 22000 КОНФИГУРАЦИЙ. КОМПЬЮТЕРЫ DELL — ЭТО **КАЧЕСТВО** BRAND-NAME, ПОЛНАЯ **СОВМЕСТИМОСТЬ**, РАСШИРЯЕМОСТЬ, **НАДЕЖНОСТЬ**, ВСЕСТОРОННЯЯ ПОДДЕРЖКА ПОЛЬЗОВАТЕЛЯ И **ТРИ ГОДА ГАРАНТИИ ПРОИЗВОДИТЕЛЯ**.

СЕМЕЙСТВО OPTIPLEX — ЛИДЕР В ИСПОЛЬЗОВАНИИ ТЕХНОЛОГИЧЕСКИХ ДОСТИЖЕНИЙ. (Enhanced IDE, флэш BIOS, поддержка технологии Plug-and-Play).

OPTIPLEX 486 — НОВЫЙ СТАНДАРТ НА РЫНКЕ ЛЕГКИХ РАБОЧИХ СТАНЦИЙ. Производительный компьютер для работы в локальной или глобальной сети. Приобретая его, Вы получаете: быстроту работы с Windows; упрощение модернизации при использовании контроллеров Enhanced IDE и флэш-BIOS; уменьшение стоимости эксплуатации.

OPTIPLEX 560 — ТЕХНОЛОГИЯ PENTIUM ПО ЦЕНЕ PC-486. Максимальное использование возможностей Pentium-60 MHz при помощи шины PCI и кэш-памяти 256 Кбайт; экономия Ваших средств через технологию наращивания мощности Over Drive; все преимущества мультимедиа.

OPTIPLEX 590 — ВЫСШАЯ ПРОИЗВОДИТЕЛЬНОСТЬ В КЛАССЕ, обеспеченная технологией шины PCI в сочетании с процессором Pentium-90 MHz. Расширенные возможности подключения до 4-х внешних устройств, в том числе HDD до 8 Гбайт, с использованием дискового интерфейса Enhanced IDE на локальной шине; экономия и надежность, обеспеченные системой энергосбережения; удобство технологии Plug-and-Play; 3 варианта корпусов.

POWEREDGE - SP, XE — БЛАГОПОЛУЧИЕ ВАШЕЙ СЕТИ. Мощные одно- и двухпроцессорные серверы. Производительность и расширяемость: Pentium до 100 MHz, до 512 MB RAM, до 144 GB HDD, до 500 пользователей в сети. Уверенность пользователя: защита данных RAID 0,1,4,5,10 уровней; система термоконтроля; встроенная диагностика; 33000 часов наработки на отказ.

LATITUDE-XP — БЕСПРЕЦЕДЕНТНАЯ СВОБОДА ПОЛЬЗОВАТЕЛЯ. Мощный notebook, обеспечивающий пользователю исключительные удобства в работе и необходимую производительность — до i486 DX-4 100 MHz, до 36 MB RAM; до 12 часов автономной работы; цветной активно-матричный дисплей 9,5"; легкость модернизации (стандарт PCMCIA, сменный HDD 340-525 в режиме Hot-Plug); широкие коммуникационные возможности.



THE REAL COMPUTER COMPANY

127238, Москва, Дмитровское шоссе 46, корпус 2
Тел: 482-4210, 482-4311. Факс: 288-9519, 482-4338
Для дилеров: 482-4144, E-Mail: IBS@IBS. MSK. SU

PERSONAL COMPUTER DIVISION



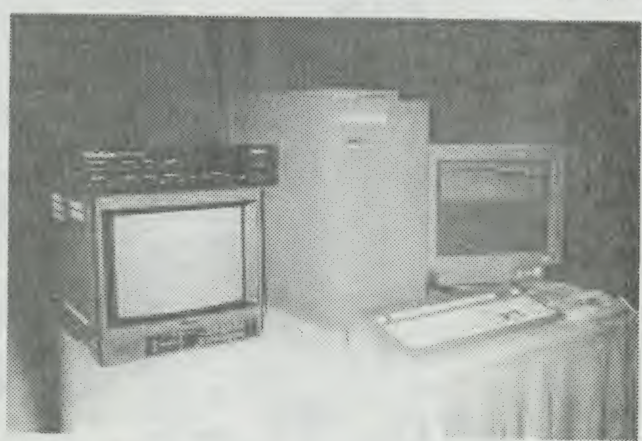
От компьютеров к системам, или Оправданный риск

Игорь Меснянкин

MACWORLD в Сан-Франциско — первая компьютерная выставка 1995 года. Года, обещающего существенно расширить присутствие Apple на мировом рынке и, будем надеяться, в России. Кроме того, это первая выставка, которую я увидел "там", и мне было интересно — действительно ли Макинтош так популярен в их американских массах. Оказалось, весьма популярен. Уже в самолете happy Mac начал демонстрировать по телевизору все свои прелести в различных рекламных роликах. А окончательно я почувствовал любовь простого народа к Макинтошам на пересадке в Нью-Йоркском аэропорту, наблюдая, как местная "тетя Маня", отпустив нам в буфете по пиву, извлекла откуда-то 230-й DUO и начала задумчиво нажимать на клавиши. "Да, Америка, — подумал я. — Что же будет на выставке?"

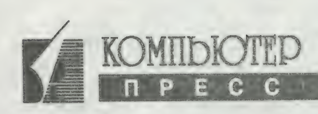
Выставка захватила меня сразу. Она подтвердила правильность политики Apple, решительно взявшей курс на выпуск компьютеров на базе RISC-процессора PowerPC. Это был рискованный шаг, но риск оказался оправданным. Уже вернувшись в Москву, мы получили сообщение, что Apple отгрузила свой миллионный PowerMacintosh спустя всего лишь 10 месяцев с момента представления первых систем на базе PowerPC. PowerPC стал сердцем нового поколения компьютеров, а PowerMacintosh — сердцем нового поколения компьютерных систем. Систем, с появлением которых пользователи получили наконец то, что им всегда недоставало, а именно — ощущение легкости и непосредственности в работе. Результаты теперь получаются настолько легко и быстро, что уже просто некогда задумываться над тем, что лучше — PowerPC или Pentium, RISC или CISC. (Забудьте о том времени, когда вы успевали выпить чашечку кофе, ожидая, пока откроется графический файл на Macintosh LC или LC II). На выставке я часто ловил себя на мысли, что вообще не замечаю компьютеров. Макинтош из просто компьютера превратился в часть системы, мощный инструмент, позволяющий выполнить вполне определенные задачи. Среди фирм-разработчиков наблюдается, по-моему, явная тенденция к концентрации сил на работе на так называемых вертикальных рынках — настольно-издательских системах, САПР, офисной автоматизации, мультимедиа и т.п. Компании предлагают не просто свои продукты, а системы, решающие конкретные задачи.

Очень внушительно выглядел стенд фирмы Radius, поглотившей не так давно другого гиганта компьютерной индустрии — компанию SuperMac. Чем же поразил Radius? Прежде всего первым компьютером Макинтош, выпущенным по лицензии Apple. Первый Macintosh-clone-компьютер "Macintosh Workstation" по внешнему виду сильно напоминал Quadra950, был подключен к системе VideoVision Telecast и монитору, а весь комплекс охранял негр-полицейский, поблескивающий идеально ровным черепом без видимых признаков растительности, что должно было, по всей вероятности, подчеркивать совершенство и законченность всей системы.



Клон Radius. Слева на мониторе — VideoVision Telecast

VideoVision Telecast — система нелинейного цифрового видеомонтажа на базе Макинтоша — одна из новинок компании. Она позволяет легко объединять видеофрагменты различных форматов (component, S-video, composite) и стандартов (NTSC, PAL, SECAM), мультипликацию, графику, записывать высококачественное звуковое сопровождение, применять спецэффекты, давая на выходе продукцию качества BetacamSP — стандарта, используемого в телевидении. И все это по цене высококачественной деки! Система обеспечивает сжатие от 3:1 до 50:1 по методу adaptive JPEG, стереозвук (16 бит, 48 кГц), прекрасные возможности аналогового и цифрового звукового ввода/вывода с двумя сбалансированными входными и четырьмя сбалансированными выходными каналами (XLR, AES/EBU, S/PDIF), поддерживает SMPTE/EBU time code. Telecast доступен как отдельный продукт



или как upgrade к уже хорошо известной системе VideoVision Studio той же фирмы и вписывается в профессиональное оборудование видеостудии так же просто, как новая видеодека. Благодаря своим характеристикам по оцифровке, сжатию изображения и синхронизации VideoVision Studio нашла применение, например, в некоторых медицинских учреждениях США, помогая врачам оттачивать технику проведения операций и постановок диагнозов с помощью компьютера в интерактивном режиме.

Для профессионалов в области издательской деятельности Radius предлагает ряд графических плат-ускорителей серии Thunder, позволяющих получать разрешение до 1600x1200 точек на дюйм и дающих возможность выбирать 16,7 млн. цветов из палитры более 1 млрд. Плата Thunder IV GX имеет 4 цифровых сигнальных процессора (DSP) и при установке на PowerMacintosh 8100/80 позволяет сэкономить до 50 минут ежедневно, если вы обрабатываете в среднем 10 изображений в день, работая в программе Photoshop, а работа в режиме CMYK в Photoshop почти не отличается по скорости от работы в режиме RGB. Radius представил также PressView 21 • T — видоизмененную PressView Display System компании SuperMac, получившей высшие оценки экспертов MacWorld и MacUser. Теперь в систему входит высококачественный 21-дюймовый монитор с трубкой, выполненной по технологии DiamondTron aperture grille (обеспечивается разрешение до 1600x1200 dpi), высокоточный колориметр (SuperMatch Display Calibrator Pro) и набор программ. С помощью колориметра измеряются цвета, производимые по отдельности красной, зеленой и синей пушками. Программное обеспечение обрабатывает эту информацию и настраивает видеоусилители этого дисплея. Таким образом, в отличие от стандартных дисплеев вы имеете возможность точно настраивать систему по цветам всякий раз, когда производите калибровку. Точка белого цвета может быть перестроена от 3000K до 9500K. Кроме того, на основе калибровочных данных строятся профили дисплея, совместимые с ведущими системами по управлению цветом: EFI EfiColor, Kodak Precision Color Management System, Apple ColorSync, Agfa FotoFlow.

С появлением все большего числа недорогих цветных принтеров и высококачественных дисплеев проблема соответствия цветов, которые получаются на экране и на отпечатках, становится все более актуальной. Radius предлагает решение в виде программы ColorComposer, позволяющей пользователю отрегулировать цвета, которые он видит на экране своего монитора, с реальными цветами, получаемыми на принтере, и даже учесть степень белизны бумаги, которая используется при печати. Эта программа поставляется с платами серии Thunder или может быть заказана отдельно.

Для тех кто идет в ногу со временем и собирается приобрести цветной копировальный аппарат серии Xerox MajestiK, компания Radius предлагает платы RISC-контроллеров Splash с программным обеспечением, превращающие Макинтош в рабочую станцию, а Xerox MajestiK — в многофункциональный сетевой цветной лазерный принтер и сканер с разрешением 400x400 dpi. Контроллеры Splash управляют всем процессом PostScript-печати, а Мак заботится о работоспособности сети и спулинге файлов. В сеть можно соединять и PC, и Маки, используя стандартное программное обеспечение Novell NetWare, Farallon PhoneNet PC, COPSTalk, Apple LocalTalk и EtherTalk.

И, конечно, если говорить о новых продуктах объединенной компании Radius, нельзя не упомянуть о семействе сетевых сублимационных принтеров ProofPositive формата A3 и A4. Новая версия RIPa 3.0 дает выигрыш в скорости печати на компьютерах PowerMacintosh на 400% по сравнению с предыдущей версией, рассчитанной на работу с Quadra.

В прошлом году компания Farallon получила высшую оценку журналов MacUser и MacWorld за свою EtherWave серию (платы, адаптеры, трансиверы, хабы), которая позволяет соединять PC, Macintosh, принтеры, портативные компьютеры в Ethernet-цепочки на витой паре. Причем можно либо подключать цепочку к одному порту хаба, либо объединять в сеть до 8 устройств без хаба. И на этой выставке Farallon порадовал своими решениями. Их новые сетевые платы для шин NuBus и PCI основаны на высокопроизводительной 3Com-технологии Fast Ethernet Parallel Tasking и обеспечивают передачу данных 10/100 мегабит в секунду. PCI-платы будут выпущены на рынок одновременно с появлением Макинтошей с шиной PCI. Хабы и мосты для Fast Ethernet будут доступны во второй половине года. О своих планах по разработке продукции для шины PCI, которая заменит NuBus во втором поколении компьютеров PowerMacintosh, объявили еще десяток фирм, и в том числе Avid Technology, Inc., Digital Equipment Corporation, Linotype-Hell AG, Radius, Inc.

Появление на российском рынке программ ArchiCAD и topCAD компании Graphisoft вызвало волну интереса к Маку как к превосходному инструменту в области САПР. Для тех, кто когда-либо занимался проектированием электронных схем в программе типа PCAD на PC, думаю, будет интересно узнать о существовании McCAD — программы электронного проектирования фирмы VAMP, дающей разработчику все преимущества использования высокопроизводительной графики Мака в сочетании с особенностями его операционной системы. Модульный принцип построения дает пользо-





Игорь Меснянкин окончил физфак МГУ в 1987 году. Работал инженером-электронщиком по разработке микропроцессорных систем, в 1990 году вернулся на физфак и в течение трех лет занимался исследовательской работой в области физики сверхпроводников. Получил грант Министерства науки на разработку сканеров на сквидах (датчиках сверхслабых магнитных полей). С 1993 года работает в компании RUI — торговым представителем, а затем менеджером отдела продаж продукции третьих фирм. В настоящее время занимается маркетинговыми исследованиями в области компьютерных решений на базе Макинтош.

вателю необходимый технический диапазон, обеспечивая контроль и гибкость, несвойственные многим системам, включая даже некоторые рабочие станции. Программа позволяет разрабатывать и редактировать цифровые и аналоговые электронные схемы, использовать стандартные библиотеки элементов и создавать свои собственные, дает возможность протестировать устройства и просмотреть временные диаграммы на этапе проектирования (с помощью цифрового симулятора или аналогового симулятора на основе стандартных алгоритмов Berkley SPICE), выполнить разводку элементов на этапе создания макета печатной платы, транслировать файлы в стандартный герберовский формат для вывода на фотоплоттер.

Бурное развитие компьютерной индустрии в последние годы и жесткая конкуренция заставляет инженерную мысль интенсивно биться, а та, в свою очередь, старается вырваться наружу и обрести достойную аппаратно-программную форму. Так, компания UMAX продемонстрировала Gemini D-16 — первый планшетный сканер с двухлинзовой системой. При использовании стандартной линзы достигается оптическое разрешение 400x400 dpi, а при переключении на другую линзу (аналог телефотолинзы в фотоаппарате) сканер дает оптическое разрешение 800x800 dpi — наивысшее для этого класса сканеров. Подход, примененный компанией, позволяет обойти ограничения, присущие технологии CCD (charge coupled device). Gemini D-16 — однопроходный 30-битный сканер с максимальным разрешением 9600x9600 dpi, он имеет встроенный интерфейс SCSI-II, 1 Мбайт кэш-памяти и является, по-видимому, самым быстродействующим планшетным сканером из доступных сегодня на рынке.

На выставке было представлено множество продуктов, связанных с использованием Макинтоша в офисе. Одним из наиболее популярных продуктов здесь оказался персональный сканер PaperPort фирмы Visioneer. Он автоматически включается, когда в него вставляют лист бумаги и через 6 секунд изображение страницы высвечивается на экране монитора. Сканер весит чуть больше килограмма и легко помещается между компьютером и клавиатурой. Вместе с программой распознавания текста он стоит около 400 долларов.

Миниатюрную видеокамеру QuickCam стоимостью 149 долларов можно было увидеть на стенде компании Connectix. Она имеет форму шара диаметром 100 мм и подключается к последовательному порту любого Макинтоша, на котором установлена программа QuickTime. QuickCam обеспечивает разрешение до 320x240 пикселов и 16 градаций серого. Камера замечательна тем, что выдает чистый цифровой сигнал, исключая таким образом искажения, связанные с преобразованиями аналогового сигнала в цифровой. Камера прекрасно подходит для проведения видеоконференций, захвата неподвижных картинок, создания небольших презентационных видеороликов.

Множество компаний предлагают электронные версии всех мыслимых и немыслимых вариантов ежедневников, баз данных деловых и личных контактов, план-календарей и т.п. Пожалуй, одно из наиболее удачных решений для делового человека представляет компания Now Software. Это пара программ Now Contact и Now Up-to-Date 3.0. Первая из них позволяет создавать базу данных контактов, создавать группы по именам, организациям и т.п. и разделять данные в сети. Программа связана с Now Up-to-Date. Так, если вам необходимо отметить в еженедельнике назначенную встречу, вы просто зацепляете мышкой нужный контакт и перетаскиваете его в план-календарь, получая в одном месте полную информацию о месте и времени встречи и все нужные координаты. Now Up-to-Date позволяет согласовывать расписания сотрудников организации по сети. Так, например, менеджер может просмотреть, какое время у его коллег уже занято какими-то делами, и принять решение о самом удобном моменте для назначения общего собрания. Здесь стоит упомянуть уже довольно популярную в России связку программ того же класса — TouchBase и DataBook Pro компании Adobe.

Надеюсь, что многое из того, о чем я упомянул, можно будет увидеть на компьютерной выставке в России. Большинство фирм стали очень серьезно рассматривать российский рынок как очень перспективный и привлекательный и собираются в той или иной форме принять участие в весенней выставке AppleExpo в Москве.

Макинтош изнутри

«Мультимедию» — в каждый дом!

Николай Иванов

▲Следующий!..

В предыдущих номерах мы препарировали Макинтоши моделей с индексом AV. Эти машины дали начало, не побоюсь громких слов, новому этапу эволюции персональных компьютеров. Впервые со времен Чарльза Бэббеджа простой обыватель мог купить в магазине бытовой прибор (вы ведь не будете спорить, что Макинтош — это бытовой прибор?), который сразу же после распаковки превращался в персональную видеостудию. Конечно, профессионалы, презрительно усмехаясь, отворачивались к своим VideoCube и Onyx. Только многие ли из нас зарабатывают на хлеб видеомонтажом? Гораздо большее число людей проводит отпуск с камерой Sony в руках. И многие из них с удовольствием потратят выходные на то, чтобы превратить заурядную съемку в простенький видеофильм с титрами и голосом за кадром. Кто сказал, что в конце второго тысячелетия мы по прежнему должны играть в "Тетрис"? Почему бы не сыграть в Lucas Films?

Правда, первая серия таких "игрушек" получилась довольно дорогостоящей. Впрочем, это не удивительно: AV-модели значительно отличались по конструкции и идеологии от более ранних Макинтошей, и Apple, естественно, предполагала покрыть расходы на их разработку за счет более высокой цены. Когда стало ясно, что машины с аудио- и видеофункциями пользуются популярностью, пришло время создать более дешевые компьютеры с похожими возможностями. Тем более, что бум на рынке домашних компьютеров в США требовал от Apple повышенного внимания к этой группе потребителей. И вот тогда-то параллельно линии мощных AV-машин на базе PowerPC компания выпустила на рынок серию 630 — группу домашних компьютеров с видеовозможностями.

630-е появились в продаже в сентябре прошлого года и сразу же обратили на себя внимание специалистов и покупателей. Некоторые обозреватели (к сожалению, не помню, кто именно) называли Performa 63x лучшими домашними компьютерами 1994 года. Я готов предположить, что эти Макинтоши играли немалую роль в финансовых достижениях Apple в конце прошлого года, когда доходы корпорации за четвертый квартал оказались на 15% выше, чем за 4 квартал 1993 года. Серия 630 уже освещалась в печати, поэтому мы не будем рассматривать ее чересчур подробно, а остановимся на наиболее интересных ее особенностях.

А вместо ▲сердца — ...

По большому счету, семейство 630 состоит из двух Макинтошей: LC 630 (процессор MC68LC040) и Quadra 630 (MC68040). Процессоры отличаются только отсутствием математического сопроцессора в MC68LC040. Все остальное в этих компьютерах абсолютно идентично (за исключением, разумеется, таблички на корпусе). В США семейство расширено за счет моделей Performa 63x, представляющих собой LC 630 в различной конфигурации, и разными наборами прилагаемых программ и компакт-дисков.

Конструкция корпуса специально приспособлена для легкой модернизации и установки плат расширения в домашних условиях.

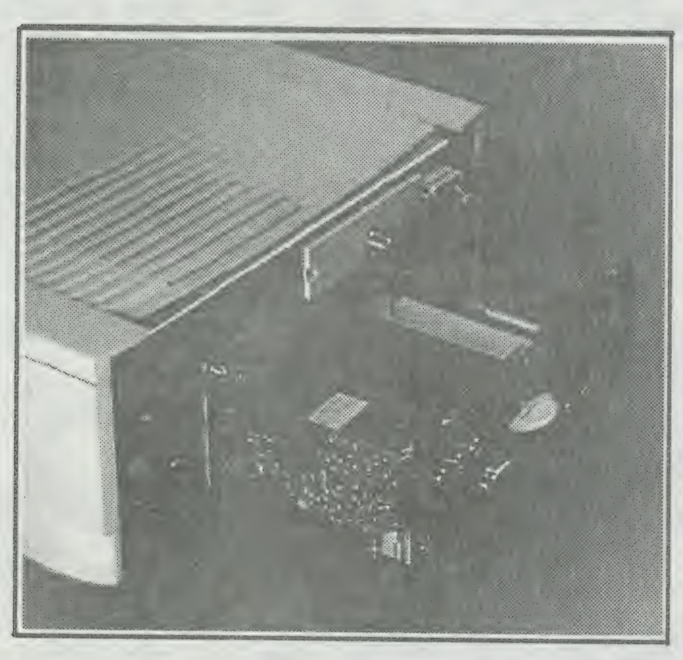


Системная плата крепится на защелках и легко выдвигается.

Некоторые технические подробности (кому не интересно — можете пропустить). Частота шины и процессора — 33/66 МГц; это значит, как легко догадаться, что процессор работает на удвоенной частоте шины. Такое обозначение частоты процессора пришло из IBM-совместимого мира, где волчьи законы конкуренции заставляют производителей вводить в заблуждение невинных покупателей, маркируя компьютеры загадочными, но очень большими числами[1]. 4 Мбайта памяти припаяно к системной плате, имеется один разъем для 72-контактного модуля памяти (SIMM) с временем доступа 70 или 80 нс; это дает при теперешнем уровне развития технологии максимум в 36 Мбайта.

[1] Например, DX2-66 означает процессор, работающий с удвоенной частотой шины, то есть 66 МГц; аналогично, DX4-100 означает, что процессор работает с... правильно, утроенной частотой, то есть 99 МГц. Внимание, вопрос: найдите на картинке цифры 3 и 9.

Процессор 68LC040 может быть сравнительно легко (нужен специальный инструмент для извлечения микросхемы) заменен на полноценный 68040, превращая LC в Quadra. Правда, цены на машины и процессоры таковы, что никакого экономического эффекта это не даст (или даже даст отрицательный). Предусмотрена возможность замены процессора на PowerPC. Для этого разработана специальная "дочерняя" плата (daughterboard), на которой помимо процессора размещен кэш второго уровня объемом 256 Кбайт. Такие платы Processor Upgrade Card уже объявлены и должны появиться в продаже уже весной. После модернизации можно переключаться между двумя процессорами; таким образом, программы и для PowerPC, и для 68 Кбайт смогут работать в "родном" (native) режиме.



Видеоконтроллер с 1 Мбайт памяти поддерживает работу с мониторами размером до 15 дюймов с разрешением 640x480 точек и глубиной цвета 16 бит (32 767 цветов).

На 15-дюймовом мониторе производства Apple поддерживается разрешение 800x600 точек, при этом глубина цвета падает до 8 бит (256 цветов). Портретные мониторы не поддерживаются из-за нестандартных частот синхронизации, хотя их традиционный размер — тоже 15 дюймов.

Серия 630 оснащена стандартными для всех Макинтошей портами — два последовательных RS422/232, совместимых с адаптерами GeoPort, один ADB для подключения клавиатуры и других устройств ввода, один SCSI, монофонический микрофонный, стереофонический линейный выход. Про два последних стоит сказать отдельно. Микрофонный вход по существу является линейным, так как рассчитан на сигнал более высокого уровня, чем стандартный микрофонный. Такой вход используется на всех моделях Макинтошей начиная с AV-машин. Для него применяется специальный микрофон PlainTalk со встроенным усилителем. Напряжение на этот усилитель подается от компьютера через удлиненный контакт модифицированного 3,5-миллиметрового разъема. Это я говорю к тому, что обычный микрофон, включая старые микрофоны Apple, не сможет нормально работать ни с одной современной моделью Макинтоша (включая LC 475 и PowerMac).

Что же касается линейного выхода, то для него в серии 630 предусмотрено два разъема — один на задней панели, как обычно, а другой — на передней. У второго есть еще и отдельный регулятор громкости — две кнопочки на рядом с гнездом для наушников. Таким образом, конструкторы пошли-таки на поводу у пользовательских масс, по неясным причинам предпочитающих слушать музыку, сидя перед компьютером, а не позади него, и регулировать громкость вручную вместо того, чтобы запускать для этого специальную программу (пульт "Звук").

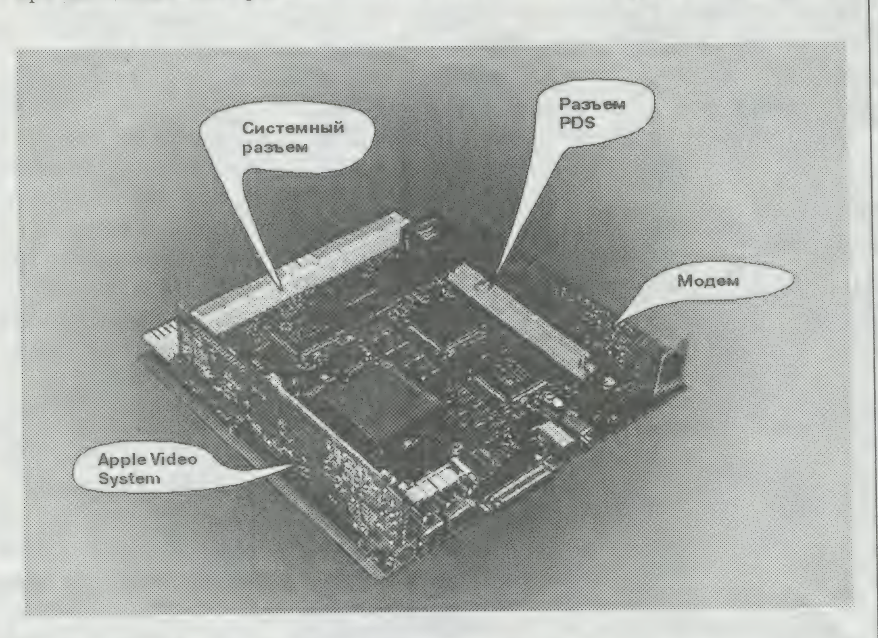
Дешевое — не значит плохое

Но самое интересное у 630-х внутри. Начать с того, что встроенный жесткий диск у них имеет интер-

фейс не SCSI, а IDE. Использование диска в стандарте IDE позволило немного снизить стоимость компьютера. Кроме того, по оценкам людей, пользовавшихся Макинтошами этой серии, новый диск работает немного быстрее традиционных SCSI. Для форматирования диска IDE прилагается специальная программа — Internal HD Formatter. Все остальные программы не замечают обмана и работают с IDE-диском как обычно. Надо отметить, что в некоторых случаях отмечается несовместимость новых дисков с традиционными дисковыми утилитами и программами архивирования при использовании режима блочной передачи. Для устранения этого недостатка Apple распространяет специальную "заплатку" — 630 SCSI Update. SCSI-интерфейс, как сказано выше, сохранился и используется, в частности, для подключения внутреннего дисковода CD-ROM.

Макинтоши серии 630 отличаются также большим разнообразием внутренних разъемов расширения. Стандартный PDS-разъем перекочевал из ранних моделей LC. Утверждается, что большинство карт расширения будут работать и с новыми Маками. Несовместимы только те карты, которые ориентируются на архитектуру процессора, например, ускорители или карты с математическими сопроцессорами 68882. Другой разъем, испытанный на модели LC 575[2], — коммуникационный. Для него в качестве карт расширения предлагаются факс-модем 14400/9600 и контроллер Ethernet.

Третий разъем — видео, специфический для этих мультимедийных машин. В него вставляется специальная плата, называемая Apple Video System, стоимостью 149 долларов (Apple Price). Эта плата представляет собой мультисистемный видеоинтерфейс, построенный на декодере Philips, и имеет композитный и S-video входы. Она, естественно, совместима с QuickTime и может быть использована с любой Маковской программой обработки видеоизображения. Примерная скорость оцифровки по результатам испытаний составляет 12 кадров в секунду при изображении размером 160x120 точек и 2-3 кадра в секунду для полноэкранной картинки (640x480 точек). Невысокая производительность связана с тем, что оцифровка производится программно. После появления процессорного модуля с PowerPC и драйвера, работающего в кодах RISC-процессора, скорость оцифровки, вероятно, возрастет в несколько раз.



Особенностью видеоинтерфейса является то, что на выходе у него получается сигнал в цветоразностном формате (YUV 4:2:2), а не RGB, как у обычных плат. Таким образом достигается более высокое качество видеоизображения, воспринимаемое зрителем. Дело в том, что человеческий глаз более чувствителен к изменению яркости, чем к изменению цвета. В цветоразностном сигнале под передачу яркости отводится 4 разряда, а под сигналы цветности — по 2 разряда, что при ограниченной разрядности полного сигнала дает субъективно более высокое качество картинки.

Встроенного видеовыхода у компьютеров серии 630 нет, однако Apple предлагает внешний конвертер Presentation System, аналогичный известному L-TV. Эта коробочка размером с пачку сигарет включается в стандартный мониторный порт и преобразует видеосигнал в форматы NTSC или PAL.

Еще один разъем предназначен для подключения телевизионного приемника, который работает только в паре с видеоплатой и продается совместно с ней под названием Apple Video/TV System за 249 долларов. Телеприемник способен принимать 181 телеканал и сигналы телетекста и выводить на экран изображение размером 320x240 точек (другие размеры изображения получаются простым увеличением размера точек). Специальная программа-"телевизор" позволяет просматривать изображение в окошке, не отвлекаясь от работы или во весь экран. Программа сравнительно легко адаптируется для приема телетекста на русском языке.

---

[2] Эта модель в России не продавалась. Представляет собой модифицированный LC 475 в одном корпусе с 14-дюймовым монитором.

Единственным недостатком телеприемника с точки зрения российского телезрителя является его несовместимость с нашим стандартом телевещания (SECAM D/K), что выливается в невозможность одновременного приема цветного телесигнала и звукового канала. Говорят (на протяжении полугода), что компания RUI готовится подписать соглашение с некой российской фирмой, которая будет адаптировать Apple Video/TV System к нашим непростым условиям.

Последним штрихом, завершающим "одомашненность" 630-х, является наличие стандартного пульта дистанционного управления, приемник которого расположен на передней панели компьютера. С помощью этого пульта можно включать Макинтош и управлять встроенным дисководом CDROM и телеприемником.

Американская цена самого "навороченного" представителя семейства — 8 Мбайт ОЗУ, диск 250 Мбайт, дисковод CD-ROM, 15-дюймовый монитор со встроенными колонками, клавиатура, мышь (ну как же без мыши), факс-модем, Apple Video/TV System, два десятка различных программ и десяток прилагаемых компакт-дисков — составляет чуть меньше 2 тысяч долларов. И главное, все это работает сразу же после распаковки, как кофемолка или двухкассетник. Практически даром покупатель получает столько удовольствия! Конечно, это не Power Mac и способности видеосистемы оставляют желать лучшего, однако, будь у вас лишние две тысячи долларов, разве вы не купили бы такое? Я бы точно купил.

Телевизор с клавиатурой

В общем, высокая технология "с человеческим лицом" рвется в каждый дом. А то ли еще будет... После появления модуля с PowerPC привлекательность Маков 630-й серии существенно возрастет. Конечно, не процессором единым... Летом этого года появятся Макинтоши со встроенной поддержкой трехмерной графики. На быстром процессоре QuickDraw 3D существенно упростит разработку 3D-приложений. Такие гранды, как 3Dlabs, auto-des-sys, Inc., Diehl Graphsoft, Inc., Electric Image, Inc., Graphisoft, Matrox Graphics, Inc., Ray Dream, Inc., Specular International, Strata Inc., Virtus Corporation заявили о готовности использовать эту технологию в своих проектах. Когда-то еще в эпоху исторического материализма комбинация из Макинтоша, принтера LaserWriter и программы PageMaker произвела революцию в издательском деле. Вам не кажется, что история готова повториться?

Компания FujiFilm Microdevices лицензировала у Apple технологию FireWire. FireWire — высокоскоростной последовательный интерфейс, основанный на стандарте IEEE P1394, обладающий пропускной способностью порядка 100 Мбайт/с. На основе FireWire FujiFilm будет разрабатывать специализированные микросхемы для построения цифровых устройств регистрации информации (фото- и видеокамеры, устройства обработки звука и т.п.). Тем самым будет удалено лишнее звено из систем обработки информации. Производительность нового интерфейса позволит передавать цифровую информацию — звук, видео — между компьютерами и периферийными устройствами в реальном времени практически без сжатия.

К тому же за последний месяц корпорация Apple объявила о новых разработках, которые впишутся в эту радужную картину как нельзя лучше. Про QuickTime VR уже довольно много написано. Теперь уже объявлено о начале продаж и условиях лицензирования инструментов разработчика "виртуальной реальности". Многие крупные разработчики, например, Broderbund Software, готовы использовать эту технологию. Представляете: Just Grandma And Me 3D...

Другая интересная вещь — Apple QuickTime Conferencing. Ничего нового в этом не было — сейчас только ленивый не разрабатывает системы для сетевых телеконференций, — если бы компания Apple не поддержала новорожденный стандарт H.320 и не собралась на его основе выпускать комплект из программы и цветной видеокамеры стоимостью не более 200 долларов. Теперь видеоконференции быстро начнут переходить из разряда забав для миллионеров в категорию потребительских товаров. И вовсе не обязательно будет ехать к дядьке в Киев, чтобы подивиться, какая у него в огороде вымахала бузина.

Вы, вероятно, слышали о специальных приставках к телевизорам (set-top boxes), предназначенных для реализации концепции интерактивного телевидения. Такие приставки изменяют однонаправленность кабельного телевещания, позволив зрителю осуществлять обратную связь с источником информации. AT&T, British Telecom, Microsoft активно экспериментируют в этой области, пытаясь придать телевизору свойства компьютерного терминала. Apple подходит к решению схожей проблемы — совмещения источника информации и устройства его отображения и обработки — с другой стороны, придавая компьютеру свойства телевизора. Видимо, пресловутая "информационная магистраль" (я бы добавил к этому штампу слово "мультимедийная") гораздо реальнее, чем это кажется на первый взгляд.

P.S. Автор выражает признательность коллективу "Радио 7" и лично Василию Стрельникову за музыкальное сопровождение, использованное при написании статьи.

e-mail: nick@jonathan.srcc.msu.su

Основы программирования на Макинтоше

Часть 7

Аркадий Морейнис

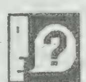
▲Менеджер диалогов

Менеджер диалогов используется для того, чтобы предупредить пользователя о какой-нибудь нештатной ситуации или запросить у него дополнительную информацию, необходимую для выполнения процесса. Макинтошевский интерфейс включает два типа окон, используемых для этой цели: предупреждения (alerts) и диалоги (dialogs). Далее в некоторых местах я буду использовать более общий термин "диалог" и для предупреждений, и собственно диалогов — надеюсь, что значение используемых терминов будет понятно из контекста. Менеджер диалогов использует менеджер окон для показа диалогов, их активации и перерисовки; менеджер управляющих элементов — для показа и обработки элементов диалога. Как правило, вам не нужно вызывать эти менеджеры напрямую, менеджер диалогов сделает это за вас.

Использование предупреждений отличается от использования диалогов. Предупреждения служат для выдачи сообщений об ошибках, а также для предупреждения пользователя о том, что его действия могут вызвать какие-либо нежелательные последствия. Предупреждение может заключаться в проигрывании звукового сигнала, показе окна предупреждения или показе окна предупреждения под звуковой сигнал (например, траурный марш). В окне предупреждений используются только следующие элементы: текст сообщения, одна или несколько кнопок и, может быть, рисунок или пиктограмма. Одна из кнопок должна быть обведена черной линией — это кнопка, которая имеет клавиатурным эквивалентом клавишу Return или Enter. Она должна соответствовать наиболее естественному действию, которое может предпринять пользователь в данной ситуации. Существуют три предопределенных типа предупреждений: "обрати внимание!" (note alert), "осторожно!" (caution alert), "нельзя!" (stop alert), отличающихся друг от друга по степени категоричности и рисуемой в левом верхнем углу пиктограмме.

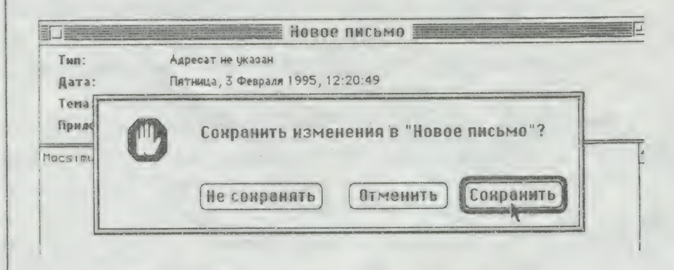
Диалоги используются для того, чтобы запросить дополнительную информацию у пользователя. Если команда из меню вызывает диалог, то после названия этой команды ставится многоточие (...). В то время как реакция пользователя на предупреждение состоит в подтверждении или отмене некоторого действия, при работе с диалогом пользователь может вводить текст, отмечать маркеры, переключатели и нажимать кнопки. В зависимости от степени свободы пользователя при работе с диалогами они разделяются по следующим категориям: модальные диало-

  "обрати внимание!"

  "осторожно!"

  "нельзя!"

ги (modal dialogs), передвигаемые модальные диалоги (movable modal dialogs), немодальные диалоги (modeless dialogs). В модальных диалогах от пользователя требуется нажать какую-либо кнопку, прежде чем процесс продолжится. После нажатия кнопки диа-

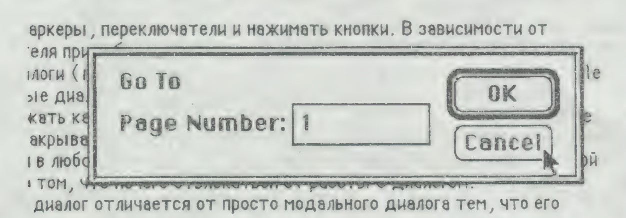


Типичное предупреждение

лог закрывается и процесс продолжается. Если же пользователь задумает щелкнуть мышкой в любом месте экрана вне диалога, то он услышит только звуковой сигнал, напоминающий ему о том, что нечего отвлекаться от работы с диалогом.

Передвигаемый модальный диалог отличается от просто модального диалога тем, что его можно двигать по экрану. При этом окно, в котором показывается

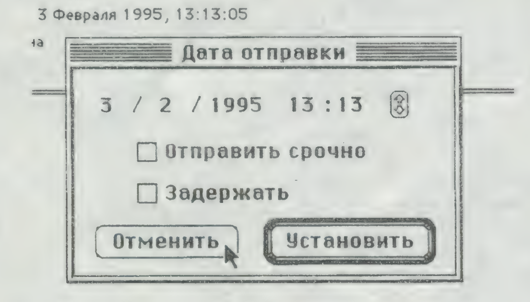
диалог, имеет титульную строку, но не имеет ни кнопки закрытия, ни кнопки кадра — диалог закрывается только в том случае, если пользователь нажимает на кнопку. Еще одно отличие передвигаемого модального диалога от просто модального заключается в том, что щелчок мышкой вне окна диалога приводит к переключению к другой программе. Правда, при возврате к исходной программе вы возвращаетесь все к тому же диалогу, на который вы должны как-то среагировать.



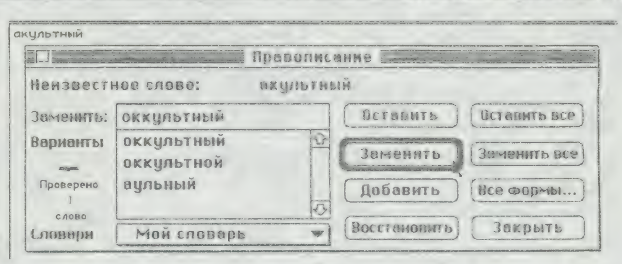
Модальный диалог

Немодальные диалоги позволяют пользователю продолжать работу в любом окне той же программы, не закрывая диалог. Окна таких диалогов снабжены титульной строкой, а также кнопкой закрытия. Такие диалоги обычно не имеют кнопки "Отменить", но имеют кнопку "Стоп", используемую для прерывания длительной операции. Закончив работу с диалогом, пользователь может его закрыть, нажав кнопку закрытия.

Элементы в диалоге могут быть активными (enabled) или неактивными (disabled). Если пользователь совершает какое-либо действие над активным элементом (щелчок мышки), сообщение об этом передается программе; если пользователь что-то делает с элементом неактивным, то программа об этом не знает. К числу неактивных элементов обычно относятся строки сообщений, пиктограммы и т.д. К числу активных — кнопки, маркеры, переключатели и т.д. Обратите внимание, что неактивный управляющий элемент и неактивный элемент диалога — это не одно и тоже. Менеджер диалогов не делает никаких визуальных различий между активными и неактивными элементами диалогов, а также не принимает во внимание, активен или неактивен управляющий элемент.



Передвигаемый модальный диалог



Немодальный диалог

Создавать диалоги можно либо на ходу, либо используя информацию, хранящуюся в ресурсах. Информация о характеристиках окна предупреждения хранится в ресурсе типа 'ALRT', информация об окне диалога — в ресурсе типа 'DLOG', а список элементов диалога — в ресурсе типа 'DITL'. Каждый элемент диалога имеет порядковый номер (начинающийся с 1), каждому элементу диалога соответствует прямоугольная область, щелчок мышкой в которой означает, что действие производится над этим элементом. В текстовой строчке — элементе диалога в ресурсе можно указывать параметры ^0, ^1, ^2, ^3. Значения этих параметров изменяются на ходу с помощью процедуры ParamText.

Для обработки модальных диалогов используется процедура ModalDialog, вызываемая после того, как программа создала и показала диалог. Если пользователь щелкнул мышкой в один из активных элементов, сообщение об этом передается в процедуру-фильтр. Процедура-фильтр может реагировать на это одним из следующих способов:

- возвращать TRUE и номер соответствующего элемента, если пользователь нажал клавишу Return или Enter;
- возвращать TRUE и номер соответствующего элемента, если пользователь нажал клавишу Escape или сочетание клавиш Command-;
- перерисовать окно диалога в ответ на сообщение о перерисовке и вернуть FALSE;
- возвращать FALSE для всех событий, которых ваша процедура-фильтр не обрабатывает.

Менеджер диалогов убирает с экрана диалог, как только пользователь нажимает любой из активных элементов, поэтому вам приходится вызывать ModalDialog в цикле до тех пор, пока пользователь не нажмет кнопку "Продолжить" или "Отменить".

Для обработки немодальных и передвигаемых модальных диалогов вы должны вызывать процедуру IsDialogEvent (либо определять любым способом, имеет ли полученное сообщение отношение к обрабатываемому диалогу) после вызова WaitNextEvent. Если полученное сообщение имеет отношение к диалогу, то вы вызываете процедуру DialogSelect, ко-

торая обрабатывает действия пользователя внутри диалога. В принципе, при обработке немодальных диалогов вы можете не использовать менеджер диалогов, а обрабатывать действия пользователя с помощью процедур менеджера окон, управляющих элементов и т.д.

Среди типов элементов, допустимых при определении диалогов, есть обычные типы элементов: кнопки, маркеры, переключатели, редактируемые строчки, нередактируемые строчки, пиктограммы, рисунки, а также любые другие управляющие элементы. Кроме этого, для создания специфического, только ему нужного элемента программист может использовать программно-задаваемый (userItem) элемент. При этом в программе с помощью процедуры SetDItem задается процедура обработки событий, соответствующая этому элементу. Эта процедура получает сообщения, имеющие отношение к данному элементу диалога, и реагирует на них соответствующим образом.

В текстовой строчке — элементе диалога в ресурсе можно указывать параметры ^0, ^1, ^2, ^3. Значения этих параметров изменяются на ходу с помощью процедуры ParamText.

```
//  указатель на диалог
typedef WindowPtr DialogPtr;
// фильтр-процедура
typedef pascal Boolean (*ModalFilterProcPtr)(DialogPtr
       theDialog, EventRecord *theEvent, short *itemHit);

// реальная структура диалога
struct DialogRecord {
 WindowRecord window;
 Handle items;
 TEHandle textH;
 short editField;
 short editOpen;
 short aDefItem;
};
// указатель на реальную структуру
typedef struct DialogRecord DialogRecord;
typedef DialogRecord *DialogPeek;

// инициализировать менеджер диалогов (передавайте nil в
       качестве параметра)
pascal void InitDialogs(ResumeProcPtr resumeProc);
// создать новый диалог
pascal DialogPtr NewDialog(void *wStorage,const Rect
       *boundsRect,ConstStr255Param  title,Boolean
       visible,short procID,WindowPtr behind,Boolean
       goAwayFlag,long refCon,
Handle itmLstHndl);
// загрузить диалог из ресурса
pascal DialogPtr GetNewDialog(short dialogID,void
       *dStorage,WindowPtr behind);
// закрыть диалог
pascal void CloseDialog(DialogPtr theDialog);
// удалить диалог из памяти
pascal void DisposDialog(DialogPtr theDialog);
// задать параметры сообщения
pascal void ParamText(ConstStr255Param
       param0,ConstStr255Param  param1,ConstStr255Param
       param2,ConstStr255Param  param3);
// обработка модальных диалогов
pascal void ModalDialog(ModalFilterProcPtr  filterProc,short
       *itemHit);
// имеет ли событие отношение к диалогу?
pascal Boolean IsDialogEvent(const EventRecord *theEvent);
// передать событие менеджеру диалогов
pascal Boolean DialogSelect(const EventRecord
       *theEvent,DialogPtr  *theDialog,
short *itemHit);
// нарисовать диалог
pascal void DrawDialog(DialogPtr theDialog);
// перерисовать часть диалога
pascal void UpdateDialog(DialogPtr  theDialog,RgnHandle
       updateRgn);
// вызвать предупреждение
pascal short Alert(short alertID,ModalFilterProcPtr
       filterProc);
pascal short StopAlert(short alertID,ModalFilterProcPtr
       filterProc);
pascal short NoteAlert(short alertID,ModalFilterProcPtr
       filterProc);
pascal short CautionAlert(short alertID,ModalFilterProcPtr
       filterProc);
// получить характеристики элемента диалога
pascal void GetDItem(DialogPtr theDialog,short itemNo,short
       *itemType,
                  Handle *item,  Rect *box);
// установить характеристики элемента диалога
pascal void SetDItem(DialogPtr theDialog,short itemNo,short
       itemType,
                  Handle item, const Rect *box);
// спрятать элемент
pascal void HideDItem(DialogPtr theDialog,short itemNo);
// показать элемент
pascal void ShowDItem(DialogPtr theDialog,short itemNo);
// выделить часть текста в строке
pascal void SelIText(DialogPtr theDialog,short itemNo,
                  short strtSel, short endSel);
// получить значение строки
pascal void GetIText(Handle item,Str255 text);
// изменить значение строки
pascal void SetIText(Handle item,ConstStr255Param text);
// на какой элемент мы попали?
pascal short FindDItem(DialogPtr theDialog,Point thePt);
// вырезать выделенный фрагмент текста
pascal void DlgCut(DialogPtr theDialog);
// вставить фрагмент текста
pascal void DlgPaste(DialogPtr theDialog);
// скопировать выделенный фрагмент текста
pascal void DlgCopy(DialogPtr theDialog);
// удалить выделенный фрагмент текста
pascal void DlgDelete(DialogPtr theDialog);
```

Macsimum
arkady@jonathan.srcc.msu.su

ПРИЗ-ОТДЫХ ЗА РУБЕЖОМ!

ВНИМАНИЕ! ЛОТЕРЕЯ ДЛЯ ЗАРЕГИСТРИРОВАННЫХ ПОЛЬЗОВАТЕЛЕЙ!

Система распознавания нового поколения

new version 2.0



Новая версия 2.0 включает новейшие достижения ведущих российских ученых и программистов, которым еще нет аналогов!

Опираясь на восьмилетние исследования в области искусственного интеллекта сегодня мы предлагаем Вам новейшую технологию ввода текста со сканера для ежедневного использования в бизнесе, издательской и научной деятельности.

Hewlett-Packard, Epson, MasElectronik комплектуют свои сканеры системой CuneiForm
Corel Corp. использует CuneiForm в программе CorelDRAW!
Inzer Corp. встроила CuneiForm в FaxLine 2.0/OCR
И тысячи людей покупают CuneiForm во всем мире!

CuneiForm 2.0 это

РАЗ:

Единственная система, распознающая любые шрифты без обучения на 10 европейских языках (включая русский) и смешанные тексты, обеспечивая при этом наилучшее (по результатам последних тестов) распознавание текста низкого качества!

ДВА:

Впервые! Благодаря научному открытию CuneiForm 2.0 распознает смешанные тексты без потери качества "основного" языка!

ТРИ:

Единственная система, сохраняющая графику (16.7 миллионов цветов) и полную копию текста (типы и начертания шрифтов, выключка, таблицы) в формате RTF, и обеспечивающая прямой вызов из популярных текстовых редакторов MS Word for Windows и Lotus AmiPro for Windows.

Вы можете купить CuneiForm рядом с Вашим домом

АЛМАТЫ
Компьютерные системы (3272) 42-5772
АСТРАХАНЬ
Комвей (8510) 33-2772
БЕЛГОРОД
КомпИнТех (07222) 2-90-04
ВЛАДИВОСТОК
Дело (4232) 25-85-11
КАЗАНЬ
НПО-Волга (8432) 75-43-06
КИЕВ
INT (044) 290-7431
Инкопартнер (044) 266-4095
ВИС Ltd. (044) 555-0796
МИНСК
НПП"Аксиома" (0172) 77-1007
МОСКВА
АГАМА (095) 443-7873
Весть (095) 115-9783
Информатик (095) 299-9904
Корт (095) 268-7478
Партия (095) 433-9330

МОСКВА
ПроСофт-М (095) 928-4714
Электон (095) 237-7102
AVICOM (095) 251-5666
Extel (095) 114-5084
Fitec (095) 433-3457
ParaGraf (095) 299-7923
R-Style (095) 316-1001
Scan Ltd. (095) 143-6641
SoftLine (095) 148-5284
SoftUnion (095) 261-9629
Tops (095) 253-6971
НОВОСИБИРСК
Софт Ателье (3832) 35-0922
Fort (3832) 28-2940
ОДЕССА
ARUS-Одесса (0452) 65-7055
САНКТ-ПЕТЕРБУРГ
СофтЮнион (812) 234-2782
Erimex (812) 242-0027
ТАШКЕНТ
Информ-Проспект (3712) 54-8858

ТОМСК
Маркетинг-Сервис (3822) 23-2706

Официальные дистрибьюторы

КИЕВ
Квазар Микро (044) 517-2675
МОСКВА
1С (095) 253-8976
Computer Mechanics (095) 332-4352
CPS (095) 930-3460
Lamport (095) 125-1101
Steepler (095) 246-1432
RSI (095) 181-2648
Радом (095) 256-6271
САНКТ-ПЕТЕРБУРГ
ПроМТ (095) 275-7887
Интек (812) 290-2200
ХАРЬКОВ
ЭлекомпТВ (0572) 30-7891

РОССИЯ, 117312, МОСКВА
ПРОСПЕКТ 60-ЛЕТ ОКТЯБРЯ, 9, ОФФ.601
ТЕЛЕФОНЫ: 135-5088, 135-4232
E:MAIL ROOT@COGTECH.MSK.SU



Каталог русскоязычных продуктов для Макинтоша

Русифицированное и оригинальное программное обеспечение компьютеров Apple

"Больше информации о Маках!" – таков лейтмотив большинства полученных нами анкет спецвыпуска "Мир Apple" (см. КомпьютерПресс № 11'94, стр. 133). И мы открываем в нашей рубрике справочно-информационный блок.

Признаюсь, я впервые столкнулся с таким делом, как анкетирование, и даже не представлял важности подобной обратной связи. Уже по прочтении первой сотни анкет чувствуешь, как зашевелились киты, на которых стояло твое представление о мире Макинтошей в России, как поворачивается твоя система координат. В одном из летних номеров мы подведем итоги анкетирования, но уже сейчас виден ряд аспектов, на которые необходимо обратить внимание. И одна из наиболее интересующих всех проблем – локализация программного обеспечения, то есть перевод программ на русский язык и, шире, адаптирование их под наши обычаи и традиции. Поэтому в качестве первой, немедленной реакции на анкеты мы начинаем печатать краткие сведения о программах для Macintosh, уже локализованных или изначально написанных по-русски. Список подготовил для КомпьютерПресс Игорь Меснянкин, эксперт RUI по комплексным решениям.

В следующих номерах мы продолжим публикацию этого "регистра", чтобы российский пользователь лучше знал текущее состояние дел. Приглашаем разработчиков и локализаторов сообщать о своих продуктах – для включения информации о них в наш регистр. И еще три коротких замечания.

Первое. В ряде анкет встречается пожелание скорейшей локализации Операционной системы Макинтош. Видимо, их авторы пока мало знакомы с Маком, и требуется небольшое пояснение. Дело в том, что несмотря на поздний приход Apple в Россию, она оказалась первой западной фирмой, которая полностью локализовала свои компьютеры. С самого начала любой легально ввезенный сюда Мак поставляется с русской клавиатурой, набором русской документации и – русифицированной вплоть до последнего системного сообщения или расширения операционной системой. (Заметим, впрочем, что последнее не всем нравится, и они предпочитают работать с американской версией, в международном пространстве терминов "Desktop", "Folder", "Icon" и "Get Info" – вместо "Стол", "Папка", "Картинка" и "Паспорт". Это тоже возможно – путем установки максимумовского Диалекта английская система становится двуязычной, не теряя при этом своего английского "лица".) С выходом новых версий и расширений системы они немедленно локализуются – с отставанием от оригинала не более месяца.

Вообще тема локализации, работы с разными языками на Маке заслуживает более подробного разговора, который мы надеемся провести уже в следующем номере.

Второе. Сразу бросается в глаза отсутствие в списке продуктов Microsoft – Word, Excel, всего набора Microsoft Office, которые из года в год занимают первые места в списках наиболее продаваемых программ для Мака. Вопрос о выпуске русских версий этих программ уже около года решается где-то в недрах Microsoft. Будем надеяться, что бюрократические колесики все же провернутся. С другой стороны, с моей точки зрения, для подавляющего числа русскоязычных пользователей Мака это совершенно не критичная проблема. Интенсивно пользуясь маковским Word уже больше двух лет, я только сейчас задумался, что он ведь, собственно говоря, не локализован! С русскими текстами он работает без всяких проблем, без каких-либо дополнительных действий с вашей стороны, как и всякая нормальная маковская программа, видит и использует все имеющиеся в системе русские шрифты, печатает, ищет, меняет размеры и так далее. (Кстати, для тех, кто не знаком с Маком: "установка шрифтов" заключается ровно в том, что вы зацепляете мышью иконку нужного шрифта или целой группы шрифтов, тащите к папке "Система" – и там отпускаете. Все. С этого мгновения любая программа, от Word и Quark до Photoshop и TouchBase, знает и может использовать эти шрифты. Чуть сложнее процесс "убирания" шрифта из системы: вам придется все же открыть папку Система, в ней папку Шрифты, и уже оттуда утащить иконки ненужных шрифтов в любое место – на стол, в корзину, в другую папку.) Собственно говоря, нелокализованность Word в том, что по-английски высвечиваются меню и команды – но по мне так проще работать, не надо заниматься обратным переводом. Более существенна локализация спеллинга, переносов и Help – внешних по сути функций, которыми я, впрочем, все равно не пользуюсь (но которые именно в качестве внешних функций, самостоятельных продуктов уже реализованы для русского языка – см. наш регистр). Наибольшим неудобством для меня был обмен русскими текстами с PC – их из-за разных кодировок приходилось передавать через текстовый ASCII-файл, теряя, естественно, при этом форматы и стили. Но говорят, в 7.5 уже появился нормальный конвертор, переводящий текст Word PC в Word Mac и обратно с полным сохранением Word-овских форматов. Примерно те же слова можно сказать и про Excel – все прекрасно работают на нем с русскими текстами.

Третье. В нашей анкете были три однотипных вопроса: "Несколько слов для журнала/ для Apple в России/для составителей спецвыпуска". Как выяснилось, тут я дал маху, опустив одно слово – надо было писать: "Несколько конструктивных слов". Увы, достаточно многие приняли формулировку за приглашение к комплиментам и вписали ответы типа "так держать", "своевременная инициатива", "любимый журнал". Спасибо за эти оценки, однако главной целью публикации анкеты была именно коррекция курса. Я думаю, у всех, даже у полностью разделяющих наше общее направление, имеются какие-то свои соображения по этому поводу, предпочтения, замечания, претензии и пожелания относительно скорости следования, мест стоянок и бортового меню. Впрочем, многие все же прислали очень и очень полезные замечания, а некоторые даже целые приложения на нескольких страницах с подробным разбором полетов и разработанными тактикой и стратегией на будущее для всех трех адресатов. Большое спасибо всем, кто заполнил и прислал – или еще пришлет – анкеты! Мы готовы – и были бы очень признательны – получить более критичные замечания и от остальных читателей – даже если анкета уже отправлена. Остаемся в контакте.

Сергей Новосельцев



▲ Интегрированные пакеты

RagTime 3

Интегрированный пакет включает: текстовый процессор, электронную таблицу, бизнес-графику, а также элементы издательских программ.

- Текстовый редактор: правильное расставление переносов на 15 языках, включая русский, автонумерация страниц и параграфов, даты и времени, поиск и замена стиля, шрифта, размера, цвета, языка, импорт файлов из MS Word, WriteNow, MacWrite, ASCII-файлов.
- Электронная таблица: до 16 000 строк и 253 колонок, 80 встроенных функций и кнопок.
- Бизнес-графика: построение дву- и трехмерных диаграмм, рамок с текстом и графикой, встроенный интерфейс со сканером, работа с 32-битным цветом.

Разработчик: MacVONK Canada Inc.
Локализация: INOVEX electronics GmbH.
Тел.: (095) 150-8840; **факс:** (095) 943-0041

Claris Works 2.1

Интегрированный пакет, включающий электронную таблицу, текстовый процессор, графику, базу данных, средства коммуникации (режим почты для 7.5). Дает возможность создавать презентации в виде слайд-фильмов и вставлять видеофрагменты в формате QuickTime. Позволяет работать с различными стилями, поддерживает UniSpell, а также словарь английского языка.

Разработчик: Claris.
Локализация: RUI IMC Apple Computer.
Доступность: любой дилер RUI IMC Apple Computer.

▲ Программы — организаторы офисной работы

Team Agenda

Помогает организовать план-календарь деятельности сотрудника компании и скоординировать действия и ресурсы всей организации.

Основные черты: графический или хронологический календарь, список дел (незавершенные дела могут автоматически переноситься на следующий день до их выполнения), текстовые или звуковые напоминания о делах, перемещение их по календарю мышкой (drag & drop), синхронизация и просмотр нескольких планов одновременно (дневной, недельный форматы, планы разных сотрудников), делегирование заданий в план другого сотрудника, при использовании System 7.5 – прием/посылка почты.

Разработчик: MacVONK Canada Inc.
Локализация: INOVEX electronics GmbH.
Тел.: (095) 150-8840; **факс:** (095) 943-0041

Полка

Проблемно-ориентированный пакет для организации безбумажного документооборота. Ведение ежедневников и телефонных книг, поиск и обмен данными по сети, голосовое сопровождение; при наличии факс-модема – отправка/прием факсов, автодозвон, электронная подпись, ведение книги организаций и переговоров. Вся информация хранится в отдельных файлах базы данных, связанных с помощью системы перекрестных ссылок.

Разработчик: СП "Интерпроект".
Тел.: (095) 258-4460; **факс:** (095) 973-2003

AgentDA, Russian

Удобная, компактная, гибкая, быстродействующая программа-календарь с автоматическим оповещением о событиях и контролем выполнения дел. По мнению MacWeek – это одна из тех программ, которые позволяют почувствовать себя опытным пользователем в считанные минуты.

Разработчик: Team Building Technologies, Inc.
Локализация: BigMac Computer Trading Comp.
Тел: (095) 235-1667, 235-5120, 235-1369
Факс: (095) 235-1369

▲ Системы учета, анализа и планирования деятельности предприятия

Суперменеджер

Программный комплекс для автоматизированного ведения бухгалтерского учета и отчетности.

Основные достоинства: модульный принцип построения, учет в рублях и любой национальной валюте, автоматический расчет курсовых разниц, полное информационное взаимодействие, дружественный интерфейс, простота в обучении и эксплуатации, гибкость к изменениям в законодательстве, учет в нескольких организациях и консолидация банков, получение отчетности в сопоставимых ценах, возможность совместной работы Макинтошей и IBM PC-совместимых компьютеров.

Разработчик: ЛАНКС
Перенос на платформу Макинтош: Macsimum
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

MacHanza A/C II (version 2.1)

Интегрированная система учета, анализа и планирования деятельности предприятия с многооконным представлением информации и технологией "клиент/сервер", работающая на компьютерах Macintosh и PowerMacintosh и в среде Microsoft Windows. Программа построена по модульному принципу, причем данные, вносимые в отдельный модуль, автоматически учитываются и обрабатываются в других. Настройка и перенастройка программы, изменение форм отчетов и документов могут быть выполнены пользователем. Основные модули: Склад, Бухгалтерия, Производство, Поставщики, Заказы поставщикам, Предложения, Заказы, Реализация, Проекты, Сервер. Заключение договора поддержки дает право на получение консультаций и новых версий.

Разработчик: Baltic Business Syst.
Локализация: Deep Apple.
Тел.: (095) 203-8233; 203-8255
Факс: (095) 203-2139

Бухгалтерский учет

Функционально законченная, гибкая, способная к расширению система учета, построенная на базе Microsoft Excel. Является единой структурой электронных таблиц/журналов и ведомостей, связанных между собой программным путем. Создана с применением бухгалтерских регистров и форм, предусмотренных Министерством финансов от 2 июля 1991 года.

Разработчик: ТОО "СП СЕПТ"; BigMac Computer Trading Company
Тел: (095) 235-1667, 235-5120, 235-1369
Факс: (095) 235-1369

▲ Системы управления базами данных и текстовые процессоры

FileMaker Pro 1.0

Система управления базами данных, которую характеризуют высокая производительность в сочетании с простотой пользования. Позволяет быстро находить информацию, группировать и анализировать данные, рассчитывать различные показатели и печатать результаты.

Основные свойства: создание неограниченного числа макетов для просмотра, ввода данных, генерации отчетов, возможность использования до 81 цвета, автоматизация типовых задач с помощью командных файлов, определяемых пользователем, и графических кнопок, ввод данных с использованием средств типа переключателей, меню, списков.

Разработчик: Claris.
Локализация: RUI IMC Apple Computer.
Доступность: любой дилер RUI IMC Apple Computer.

WorldWrite with SpellChecker

Текстовый процессор с возможностью проверки правописания. Позволяет работать с несколькими колонками, использовать несколько слоев, различные стили, поддерживает форматы PICT, EPS, TIFF, поддерживает 2 метода цветоделения (Spot и Process), дает возможность создавать таблицы, выполнять масштабирование, осуществлять поиск и замену.

Разработчик: WorldSoftware Corporation.
Локализация: Macsimum Ltd.
Доступность: любой дилер RUI IMC Apple Computer.

МаксимумТекст

Редактор, дополняющий существующие настольно-издательские системы и текстовые процессоры инструментальными средствами для работы с русскими текстами.

Основные свойства: проверка орфографии по словарю, включающему 100 000 основ с предложением вариантов исправления, возможность занесения новых слов в словари, проверка типографской корректности пунктуации, интеллектуальный поиск/замена, расстановка переносов, словарь исключений позиций переноса в словах, неограниченный размер редактируемого файла, средства редактирования текста (смена регистра, алфавита и т.п.), простейшие средства редактирования документа.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

WriteNow 3.0, Russian with Webster's Collegiate Thesaurus

Наиболее простой в освоении текстовый редактор, по возможностям немногим уступающий MS Word. Устойчиво и быстро работает под всеми ОС как на Macintosh, так и на PowerMacintosh. Оснащен английским словарем и тезаурусом. Имеет фильтры для работы с файлами форматов MacWrite, MS Word, MS Works, WordPerfect, RTF.

Разработчик: T/Maker Company
Локализация: BigMac Computer Trading Company
Тел: (095) 235-1667, 235-5120, 235-1369
Факс: (095) 235-1369

▲ Коммуникационные пакеты

FirstClass

Предназначена для использования в качестве как внутриофисной электронной почты, так и для организации BBS для обмена информацией по сети и телефонным линиям, проведения электронных конференций для группы пользователей.

Эта программа позволяет пользователям MacOS и Windows сэкономить деньги и не покупать сложный файловый сервер или рутер, необходимый для работы других мультиплатформных коммуникационных пакетов. Версия клиента распространяется бесплатно.

Разработчик: SoftArc Inc.
Локализация: Колибри.
Тел.: (095) 306-1772; **BBS:** 915-7259

Macsimum Mail 1.0

Система электронной почты с выходом в систему Relcom и другие сети, поддерживающие протокол UUCP.

Основные свойства: простой интерфейс, возможность орфографического контроля русскоязычных текстов, ведение адресной книги и списка тем, пересылка почты в заданное время, возможность использования в качестве внутриофисной почтовой системы, автоматическая отправка и прием текста в фоновом режиме, пересылка русскоязычных текстов между Macintosh, MS-DOS/Windows, UNIX, а также двоичных файлов в форматах BinHex и UUencode, наличие макроязыка.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

Talker

Пакет программ речевой связи, использующий микрофон и динамик вашего компьютера Макинтош для переговоров с другими пользователями по сети AppleTalk. Вы можете пользоваться речевой связью как на коротких расстояниях в локальных сетях, так и на длинных расстояниях в глобальных сетях, а также по радиосвязи.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

▲ Программы для издательских систем

PageMaker 5.0

Позволяет комбинировать текст, сканированное изображение и иллюстрации, создавать длинные документы с автоматическим перетеканием текста по последовательности страниц. Дает возможность поворачивать текст с точностью 0.01 градуса, выполнять цветоделение, просматривать и работать одновременно с несколькими открытыми публикациями и переносить объекты из одной в другую с помощью мыши; часто используемые объекты можно хранить в библиотеке. Встроенный текстовый редактор с возможностью поиска и замены. Система PANOSE обеспечивает обмен шрифтами между платформами. Большое количество фильтров импорта/экспорта.

Разработчик: Adobe Systems, Inc.
Локализация: Macsimum Ltd.
Доступность: любой дилер RUI IMC Apple Computer.

Quark XPress 3.3.1

Профессиональная электронная издательская система, сочетает точность, мощь и простоту в создании широкого круга публикаций. Библиотека позволяет хранить часто используемые картинки и текст. Программа разрешает применять размеры шрифтов от 2 до 720 точек с шагом 0.001, трекинг и кернинг с допуском 0.0005, повороты с шагом 0.001 градус. Quark XPress 3.3.1 позволяет импортировать и редактировать все популярные графические форматы. Включает множество функций текстового процессора.

Разработчик: Quark.
Локализация: Intermicro.
Тел.: (095) 264-2865; 264-2853
Факс: (095) 267-3400, 200-2238

Арсенал

Система управления издательским процессом, полезна при подготовке периодических изданий большого объема, помогает ведущим номера контролировать график прохождения по технологическим операциям и готовность всех входящих в номер материалов. Обеспечивает регистрацию материалов всех типов и их распределение по полосам издания, слежение за соблюдением технологии обработки материалов всех типов, объемом материалов большинства типов, слежение за сроками сдачи материалов, подготовку материалов непосредственно в номер или через загон, ведение архива номеров издания, ведение журнала обработки каждого материала, поддержку подготовки нескольких изданий.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

Non-Breakable!

Внешняя команда программы QuarkXPress, улучшающая качество верстки с учетом технико-орфографических требований к переносу строк в русском типографском наборе.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

▲ Программы распознавания текста и проверки правописания

MacTiger

Система распознавания текстов. Преобразует информацию, считанную сканером с документа, в текстовый файл, готовый для дальнейшей обработки на компьютере. Распознает 10 языков, включая русский.

Разработчик: Cognitive Technologies Ltd.
Тел.: (095) 135-5088; **факс:** (095) 135-4232

UniSpell

Программа проверки русского правописания и расстановки переносов для ряда популярных программ, в том числе Microsoft Word, Excel, FoxPro, QuarkXPress, ClarisWorks, Adobe PageMaker. Включает проверку правописания по словарю (100 000 основ), возможность занесения новых слов, словарь исключений позиций переноса в словах, проверку типографской корректности пунктуации.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

▲ Утилиты

Norton Utilities 3.0

Программа обнаружения и устранения проблем, связанных с потерей данных. Выдает предупреждения, когда следует выполнить операции по сохранению данных, такие как сохранение резервной копии или оптимизации дискового пространства. При обнаружении неполадок Norton Disk Doctor проводит диагностику и устранение неисправностей. В данную версию включена возможность выполнения резервного копирования.

Разработчик: Symantec.
Локализация: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

Connectix PowerBook Utilities 2.0

Набор утилит для увеличения производительности и удобства работы на компьютерах PowerBook. Продлевает срок службы батарей, автоматически включает режим сохранения экрана, переводит компьютер в режим "уснуть"/"разбудить", дает возможность программировать "горячие" клавиши, быстро найти курсор на экране, а также выполняет ряд других полезных функций.

Разработчик: Connectix Corp.
Локализация: RUI IMC Apple Computer.
Доступность: любой дилер RUI IMC Apple Computer.

Диалект

Программа, позволяющая использовать с англоязычной версией системного программного обеспечения ресурсы другого языка. В настоящее время поддерживает русский, башкирский и татарский языки.

Основные свойства: переключение клавиатур, использующих специальные символы разных языков, например кириллицу; использование системного, программного и непропорционального шрифтов, включающих специальные символы соответствующего языка (типа Chicago, Geneva и Monaco); применение форматов даты, времени, чисел и денежных единиц, принятых в данной стране, использование процедур сортировки, учитывающих расположение букв соответствующего алфавита.

Разработчик: Macsimum Ltd.
Тел.: (095) 939-2327; 939-2471
Факс: (095) 939-1022

Конвертор v1.1

Выполняет перекодировку символьной информации, которая находится в файлах разного типа в соответствии со стандартами кодировки кириллицы, принятыми в разных системах. Поддерживается тип кодировки DOS или Windows. Обеспечивается пакетное преобразование файлов в форматах ASCII, RTF, DBF и Лексикон.

Разработчик: BigMac Computer Trading Company
Тел: (095) 235-1667, 235-5120, 235-1369
Факс: (095) 235-1369

Ваши дополнения к каталогу присылайте
Меснянкину Игорю, тел.: (095) 978-2000; 978-3869
e-mail: RUI.SOLUTION@applelink.apple.com
или **Новосельцеву Сергею**,
тел.: (095) 151-6790
e-mail next@ipian15.ipian.msk.su

Оборудование PCME

для эффективного использования каналов связи

Алексей Любимов

Особенности развития современных средств связи в значительной мере определяются быстро растущим объемом передаваемого трафика, появлением новых сетевых служб и услуг, предоставляемых пользователям, изменением соотношения различных видов трафика. В этих условиях поставщики сетевых услуг должны чутко реагировать на рост потребностей пользователей в объеме передач, увеличивая количество и пропускную способность каналов связи, а также повышая эффективность их использования. В настоящее время особенно важно последнее, так как стоимость каналов связи снижается значительно медленнее, чем стоимость обработки информации.

Передача данных методом коммутации пакетов

Для повышения эффективности использования пропускной способности канала связи при передаче цифровых данных в последнее десятилетие интенсивно проводился переход от метода коммутации каналов к методу коммутации пакетов. При коммутации каналов по запросу пользователя осуществляется соединение оконечного оборудования источника и приемника для передачи информации, то есть организуется физический тракт передачи — канал связи — и обеспечивается его монопольное использование до тех пор, пока соединение не разъединяется. Такое использование канала связи выгодно при непрерывной передаче больших объемов трафика, но при работе в диалоговом режиме (например, при передаче речи) во время пауз в данном канале не передается никакая информация, что является причиной низкой эффективности его использования. В случае коммутации пакетов сообщение источника разрезается на пакеты, каждый из которых снабжается заголовком, содержащим адрес и другую управляющую информацию. Затем эти пакеты пересылаются по сети, используя определенные маршруты движения. При передаче пакета через каждый канал связи между двумя узлами сети к нему добавляются соответствующий заголовок канального уровня и поле проверочной комбинации, действующие только в пределах данного канала. Пакеты с такими добавками называются кадрами. В узле назначения все пакеты собираются в исходное сообщение. Пакеты могут передаваться в дейтаграммном режиме, когда каждый пакет поступает от источника к получателю независимо от всех предыдущих и последующих пакетов того же сеанса связи. Порядок приема пакетов получателем не всегда совпадает с порядком их передачи, так как дейтаграммы могут следовать к получателю разными маршрутами. Другой вариант передачи пакетов — использование виртуального канала, то есть имеющего идентификатор логического канала, организуемого между двумя устройствами, желающими установить связь, по которому передаются пакеты, следующие в порядке их отправки. В одном физическом канале связи, как правило, передаются совместно пакеты, принадлежащие множеству различных виртуальных каналов. Виртуальные каналы могут быть постоянными и организуемыми только на время одного сеанса связи.

Цифровая интерполяция речи

При передаче телефонии улучшение эффективности использования каналов связи до недавнего времени было связано с применением цифровой интерполяции речи (ЦИР), процедуры, предоставляющей канал связи в паузах речевого сигнала для передачи информации других абонентов. Аппаратура, использующая ЦИР, называется DCME (Digital Circuit Multiplication Equipment). DCME очень эффективна при ее установке у двух окончаний канала связи (подводного трансокеанского кабеля или спутниковой линии) со скоростью передачи, называемой первичной (2048 или 1544 Кбит/с). В точке передачи на вход DCME поступают несколько (обычно восемь) каналов с первичной скоростью, которые уплотняются в один за счет использования ЦИР и низкоскоростного кодирования как речи, так и различных сигналов речевого диапазона (СРД). В точке приема осу-

ществляется обратное преобразование, разуплотняющее принятый групповой сигнал. Громоздкая система синхронизации DCME снижает гибкость в применении этой аппаратуры, препятствуя передаче информации по сети различным абонентам в уплотненном виде. Другим недостатком цифровой интерполяции речи является возможная потеря при перегрузке части речевого сигнала в начале активных участков речи. В результате в начале слов могут срезаться короткие фонемы, соответствующие взрывным звукам, — к, п, б, д и т.д., длительность которых не превышает 20-25 мс, что влечет за собой снижение разборчивости речи.

Уплотнение трафика системой PCMS

Учитывая вышеуказанные недостатки DCME, а также двигаясь в направлении интеграции служб (ибо, как показывают теоретические исследования, эксперименты и конкретные разработки, именно цифровая сеть с интеграцией служб способна обеспечить наиболее эффективное использование основных ресурсов сети, унифицировать программные и аппаратные средства, существенно расширить перечень возможных режимов доставки, обеспечить гибкость, живучесть и высокую экономическую эффективность сети связи), сотрудники AT&T Bell Laboratories в конце 80-х годов начали исследования методов совместной передачи трафика различных типов (речи, данных, факсимильных сообщений, видеоизображений и т.д.) в пакетном виде. Одновременно на основе этих исследований Международным консультативным комитетом по телефонии и телеграфии (МККТТ) проводились работы по стандартизации оборудования мультиплексирования с коммутацией пакетов PCME (Packetized Circuit Multiplication Equipment). В результате в 1990 году увидела свет Рекомендация МККТТ G.764, содержащая описание передачи речи в пакетном режиме и протокола пакетной передачи речи, а в 1992 году — Рекомендация МККТТ G.765, стандартизирующая оборудование мультиплексирования с коммутацией пакетов. В соответствии с требованиями этих Рекомендаций два и более комплекта оборудования PCME образуют систему мультиплексирования с коммутацией пакетов PCMS (Packetized Circuit Multiplication System), на вход которой поступает трафик различных типов, формируется в пакеты с использованием аналого-цифрового преобразования и низкоскоростного кодирования, передается в пакетном виде через сеть связи или по отдельному каналу связи, обеспечивая тем самым более эффективное использование каналов связи, чем традиционная коммутация каналов, и восстанавливается в исходном виде в точке приема. Между коммутируемыми каналами, подключенными к разным комплектам оборудования PCME, организуются постоянные виртуальные каналы. Тем самым обеспечивается логическая непрерывность физических коммутируемых каналов. Для пользователя сети участие систем PCMS в передаче информации практически незаметно, и весь путь передачи информации выглядит как один физический канал (обычно с некоторой транспортной задержкой). Основное достоинство PCMS — это гибкость. Пакеты могут передаваться не только по отдельному каналу, но и через сети, используя несколько промежуточных узлов коммутации и линий связи. Имеется возможность осуществления гибкой перемаршрутизации трафика в зависимости от исправности и загруженности отдельных линий связи. Один абонент с помощью систем PCMS способен одновременно передавать информацию нескольким адресатам. Одно излучение спутника может быть принято станциями разных стран, так что каждая из них воспринимает адресованные только ей пакеты. В результате цена пользования спутниковой линией связи делится между несколькими арендаторами, что влечет за собой удешевление услуг связи. Оборудование PCME может коммутировать пакеты на несколько своих выходов, имеющих различные скорости, в том числе и меньшие первичной. Это важно для стран с небольшими потребностями в объеме трафика и стало особенно ценным после того, как консорциум IN-TELSAT предусмотрел специальный более низкий тариф для скоростей 512 Кбит/с. Рассмотрим подробнее особенности передачи системами PCMS трафика различного вида.

Передача речи и сигналов речевого диапазона

Передача речи в пакетном режиме имеет следующие особенности: жесткие требования по времени доставки (задержка выше определенного предела может приводить на приемном конце к неразборчивости речи и к таким явлениям, как эхо); жесткие требования по девиации задержки (сращивание отдельных частей слов, разделенных перерывом, или значительное увеличение продолжительности перерыва недопустимы); широкий диапазон возможных значений скоростей передачи; большая смысловая и сигнальная избыточность; высокая доля пауз. Все эти особенности учтены в протоколе PVP (Packet Voice Protocol), регламентирующем передачу речи и сигналов речевого диапазона. Согласно требованиям этого протокола поступающие в точку передачи с частотой 8 кГц отсчеты речи или СРД, кодированные одним из нескольких возможных методов, обычно перекодируются в соответствии с алгоритмами адаптивной дифференциальной импульсно-кодовой модуляцией (АДИКМ, Рекомендации МККТТ G.726 или G.727) или ИКМ (Рекомендация G.711) и формируются в пакеты по 128 отсчетов в каждом, что эквивалентно длительности участка речевого сигнала 16 мс. Формирование пакетов производится таким образом, что наиболее значащие биты всех отсчетов объединяются в первый 128-битовый блок информационного пакета, следующие по значимости биты — в следующий 128-битовый блок и т.д., и наконец, наименее значащие биты всех отсчетов образуют последний блок пакета. Информационные пакеты, принадлежащие одному абоненту, передаются по отдельному виртуальному каналу. Циклическому контролю на наличие ошибок в данном случае подвергается только заголовок кадра, а не весь кадр, так как речь и сигналы речевого диапазона менее чувствительны к отдельным сбоям, чем к задержкам или потерям пакетов. Общая задержка пакета в сети не должна превышать 400 мс. Для уменьшения девиации задержки пакетов предусмотрена следующая процедура. Время ожидания пакета (TS) в очередях промежуточных узлов отмечается в его заголовке. Если в точке приема окажется, что TS меньше, чем заданное контрольное время T, то пакет задерживается на время, равное разности между контрольным временем T и временем ожидания TS. Если же TS превышает величину контрольного времени T, то пакет должен быть уничтожен. Таким образом, величина контрольного времени является компромиссом между чрезвычайно большой задержкой и большим количеством уничтоженных пакетов. Обычно она не превышает 70 мс. При формировании TS не учитывается время распространения сигнала по связным каналам (для спутниковых линий связи земля—спутник—земля оно достигает 260 мс). В национальных сетях вышеуказанная процедура уменьшения девиации задержки пакетов в точке приема может быть заменена фиксированной задержкой первого пакета каждого активного участка речи в случае передачи речи или задержкой первого пакета данного сеанса связи в случае передачи сигналов речевого диапазона. Преимущество этого метода заключается в упрощении процедур обработки пакетов промежуточными узлами, но величина общей задержки в данном случае варьируется в большей степени, поэтому в таких системах пакетам протокола PVP назначается наивысший приоритет. Протокол PVP предусматривает возможность снижения перегрузок, которые могут возникать в промежуточных узлах связи за счет уменьшения размеров пакетов путем удаления последних блоков с наименее значащими битами. Ухудшение качества обслуживания в этом случае будет приемлемым. Количество блоков пакета, которые могут быть уничтожены при перегрузке, индицируется в его заголовке. Детектор речи позволяет не передавать пакеты во время речевых пауз. В это время в точке приема воспроизводится белый шум, уровень которого измерен в точке передачи и не превосходит 1 дБ. Телефонный разговор — это, как правило, диалог. В то время, когда один из собеседников говорит, другой обычно молчит, и наоборот. Даже если говорит один человек, в его речи имеются паузы между фразами, словами, отдельными звуками. В результате, эффективность

1	2	3	4	5	6	7	8	
0	0	Адрес						октет 1
1	Адрес							октет 2
1	1	1	1	0	1	1	1	октет 3
0	0	1	0	0	0	1	0	октет 4
C1	C2	R	R	M1	M2	R	R	октет 5
Время задержки пакета в сети								октет 6
Тип кодирования					R	R	M	октет 7
Поле шума				Номер пакета				октет 8
Неуничтожаемые блоки								
Уничтожаемые блоки								
Последовательность проверки 2 октета								

Рис. 1. Формат информационного кадра протокола PVP

использования коммутируемого телефонного канала составляет 30-40%. Таким образом, переход от традиционной коммутации каналов к передаче речи методом коммутации пакетов позволяет повысить эффективность использования канала связи почти втрое. Формат информационного кадра протокола PVP представлен на рис. 1. Как видно из рисунка, первые 2 бита первого октета кадра равны нулю. Первый бит второго октета равен единице. Остальные 13 бит первых двух октетов группируются, образуя единый идентификатор соединения канального уровня или адрес канального уровня. Третий октет имеет значение 11110111. Последние два октета кадра — последовательность циклического контроля заголовка (а не всего кадра). Алгоритм вычисления контрольной суммы аналогичен алгоритму стандарта ISO 3309. Первые три и последние два октета кадра принадлежат канальному уровню. Октеты с четвертого по восьмой являются заголовком пакетного уровня. Четвертый октет выполняет функцию дискриминатора протокола и имеет значение 00100010. Пятый октет является индикатором аннулирования блока. Его первый и второй биты указывают на число блоков, которые могут быть отброшены на любом промежуточном узле сети (первый бит — младший, второй — старший). Биты пятый и шестой указывают на общее число удаляемых блоков при первоначальном формировании пакета на исходящем конце (пятый бит — младший, шестой — старший). Биты третий, четвертый, седьмой и восьмой пятого октета зарезервированы для будущего использования и имеют нулевые значения. Шестой октет информационного кадра содержит значение времени задержки пакета в очередях сети в миллисекундах. Первые 5 бит октета 7 определяют тип кодирования речевых отсчетов в точке передачи в соответствии с табл. 1. Согласно Рекомендации G.764 восьмой бит этого октета должен принимать значение, равное 1, для всех пакетов, кроме последнего пакета речевой группы. Остальные биты октета 7 зарезервированы для будущего использования и имеют значение, равное нулю. Первые четыре бита восьмого октета содержат поле шума, указывающее на уровень фонового шума в точке передачи во время пауз. Старшие 4 бита используются для нумерации пакетов в точке передачи от 1 до 15. Если номер пакета в последовательности равен нулю, то это свидетельствует о начале речевой группы и необходимости сброса в исходное состояние декодера АДИКМ. Начиная с девятого октета передается информационное поле протокола PVP. Следует отметить, что передача речи в пакетном виде еще до появления соответствующих стандартов была успешно опробована в частных телефонных сетях, в которых распределенные учрежденческие телефонные станции связаны арендованными линиями типа Т1 (1544 Кбит/с). Это позволило обеспечить значительную выгоду от снижения потребной скорости передачи для каждого речевого канала и позволило более эффективно использовать линии связи Т1.

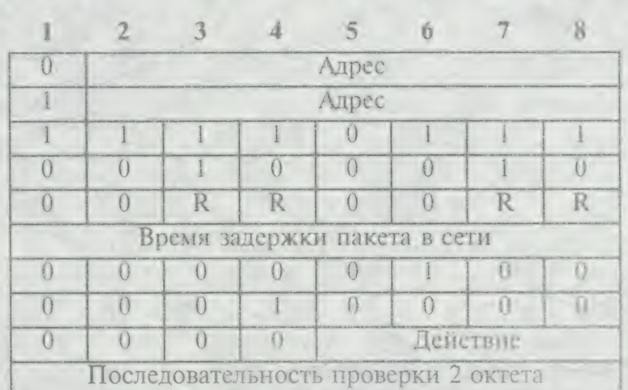
Таблица 1. Типы кодирования речевых отсчетов

Номер бита					Тип кодирования
1	2	3	4	5	
0	0	0	0	0	8 бит на отсчет
1	0	0	0	0	1 бит на отсчет
0	1	0	0	0	2 бита на отсчет
1	1	0	0	0	3 бита на отсчет
0	0	1	0	0	4 бита на отсчет
1	0	1	0	0	5 бит на отсчет
0	1	1	0	0	6 бит на отсчет
1	1	1	0	0	7 бит на отсчет
0	0	0	1	0	8-разрядная ИКМ по закону А
1	0	0	1	0	8-разрядная ИКМ по закону М
0	1	0	1	0	2 бита на отсчет АДИКМ G.726
1	1	0	1	0	3 бита на отсчет АДИКМ G.726
0	0	1	1	0	4 бита на отсчет АДИКМ G.726
1	0	1	1	0	5 бит на отсчет АДИКМ G.726
0	0	1	0	1	(4, 2) АДИКМ G.727
1	0	1	0	1	(5, 2) АДИКМ G.727
0	0	0	1	1	(8, 6) АДИКМ G.722

Передача сообщений факсимильной аппаратуры группы 3

МККТТ подразделяет факсимильную аппаратуру общего назначения на четыре группы. Группа 1 — аппаратура, использующая модуляцию с двумя боковыми полосами без каких-либо мер по сжатию ширины полосы передаваемого сигнала и пригодная для передачи страницы формата А4 (210x297 мм) с разрешающей способностью 4 линии на миллиметр по телефонной сети за 6 мин. Группа 2 — аппаратура, использующая методы сжатия ширины полосы (частичное подавление боковой полосы) и обеспечивающая передачу страницы в пределах 3 мин. Группа 3 — аппаратура, которая содержит средства снижения избыточной информации в сигнале документа до процесса модуляции и которая способна обеспечить время передачи документа со стандартным машинописным текстом формата А4 по телефонной сети в течение

1 мин. Группа 4 — аппаратура, содержащая средства снижения избыточной информации в сигнале документа до передачи и использующая процедуры, приемлемые для сети передачи данных (ПД) общего пользования. Хотя на сегодняшний день аппаратура групп 1 и 2 почти не используется, информационные сигналы этого типа могут передаваться через систему PCMS с помощью протокола PVP как сигналы речевого диапазона. Сигналы факсимильной аппаратуры группы 4, предназначенной для передачи информации главным образом в цифровом виде по сетям ПД или цифровым сетям с интеграцией служб (ЦСИС), могут передаваться системами PCMS с высокой эффективностью в качестве цифровых данных так, как описано в следующем пункте. Для повышения эффективности использования пропускной способности канала связи при передаче сигналов факсимильной аппаратуры группы 3, имеющей широкое распространение в настоящее время, специально для этого класса сигналов Рекомендацией G.765 предусмотрен протокол FADCOMP (Facsimile Demodulation and Compression Protocol). В соответствии с требованиями этого протокола все сигналы установления связи, определяемые Рекомендацией T.30 и модулированные в соответствии с Рекомендацией V.21, передаются также как и речь с помощью протокола PVP (тракт PVP), а сигналы факсимильного изображения, передаваемые согласно Рекомендациям МККТТ V.29 и V.27 ter, демодулируются, дескремблируются в точке передачи, разбиваются на пакеты и распространяются через сеть связи в цифровом виде с последующим обратным цифро-аналоговым преобразованием в точке приема (тракт FADCOMP). То есть, в зависимости от фазы сеанса связи информация, поступающая из коммутируемого цифрового телефонного канала, направляется в высокоскоростной канал связи либо через тракт PVP, либо через тракт FADCOMP. Протокол FADCOMP предусматривает пятиуровневую модель, в состав которой входят физический и канальный уровни, такие же как в PVP, немного отличный от описанного PVP пакетный уровень, а также еще два уровня: модуляционный и сеансовый (Call-State level). Модуляционный уровень обеспечивает демодуляцию и дескремблирование факсимильных постраничных данных в точке передачи и осуществляет процесс их вторичного скремблирования и вторичной модуляции в точке приема. На сеансовый уровень — общий для точек передачи и приема — возложены функции распознавания начала сеанса связи, определения роли оконечного узла в данном сеансе (приемник или передатчик), распознавания сигналов протокола T.30, направления трафика из коммутируемого канала в тракт PVP или FADCOMP в точке передачи и из тракта PVP или FADCOMP в коммутируемый цифровой телефонный канал в точке приема. FADCOMP предусматривает три типа пакетов: пакеты индикации факсимильных характеристик (Facsimile Capability Indication Packet), пакеты начала факсимильного потока (Facsimile Spurt Header Packet) и информационные пакеты факсимильной страницы (Facsimile Page Information Packet). В пакетах индикации факсимильных характеристик содержится информация о скорости факсимильной передачи и виде используемой модуляции. Пакеты начала факсимильного потока индицируют начало или прекращение генерации тестовой последовательности или тонов защиты от эха частотой 1700 либо 1800 Гц. Размер информационного поля пакета факсимильной страницы соответствует длительности 20 мс воспроизведения факсимильного сигнала на исходной скорости (2400, 4800, 7200 или 9600 бит/с) и не должен превышать 482 октетов. При этом в каждый байт пакета помещаются два квадбита (трибита, дибита) демодулированного сигнала (остальные биты байта имеют нулевое значение). Если последний байт информационного поля пакета получился неполным, то он дополняется нулями. Об этом делается отметка в заголовке пакета. Пакеты всех типов, в том числе и вышедшие из тракта PVP, передаются по одному постоянному виртуальному каналу. Циклической проверке, так же как и при передаче речи, подвергается только заголовок кадра, а не весь кадр. На рис. 2 и 3 представлены форматы кадров начала факсимильного потока и информационных кадров факсимильной страницы. Октеты с первого по четвертый, а также шестой и два последних октета кадров протокола FADCOMP имеют такие же значения, что и соответствующие октеты информационных кадров про-



1	2	3	4	5	6	7	8	
0	Адрес							октет 1
1	Адрес							октет 2
1	1	1	1	0	1	1	1	октет 3
0	0	1	0	0	0	1	0	октет 4
0	0	R	R	0	0	R	R	октет 5
Время задержки пакета в сети								октет 6
0	0	0	0	0	1	0	0	октет 7
0	0	0	1	0	0	0	0	октет 8
0	0	0	0	Действие				октет 9
Последовательность проверки 2 октета								октеты 10 и 11

Рис. 2. Формат кадра начала факсимильного потока протокола FADCOMP

Таблица 2. Кодирование требуемых протоколом FADCOMP действий

Код	Требуемые действия
1000	Генерация настроечной последовательности
0100	Сброс
1100	Начало генерации тона защиты от эха частотой 1700 Гц
0010	Начало генерации тона защиты от эха частотой 1800 Гц
1010	Конец генерации тона защиты от эха
1111	Никаких действий не требуется

1	2	3	4	5	6	7	8	
0	Адрес							октет 1
1	Адрес							октет 2
1	1	1	1	0	1	1	1	октет 3
0	0	1	0	0	0	1	0	октет 4
0	0	R	R	0	0	R	R	октет 5
Время задержки пакета в сети								октет 6
1	0	0	0	0	1	0	M	октет 7
Число бит			1	Номер пакета				октет 8
Информационное поле факсимильной страницы								
Последовательность проверки 2 октета								

Рис. 3. Формат информационного кадра факсимильной страницы протокола FADCOMP

токола PVP. Значение пятого октета кадров протокола FADCOMP нулевое. В кадрах начала факсимильного потока значения седьмого и восьмого октетов равны соответственно 00000100 и 00010000. Младшие четыре бита девятого октета зарезервированы для будущего использования и имеют нулевое значение, старшие — используются для обозначения в соответствии с табл. 2 действия, которое должно быть выполнено в точке приема

после получения кадра начала факсимильного потока. Первые семь бит седьмого октета информационного кадра факсимильной страницы имеют значение, равное 1000010. Восьмой бит принимает значение, равное 1, во всех информационных кадрах, использующихся для передачи факсимильной страницы, кроме последнего, где он равен нулю. Первые три бита восьмого октета информационного кадра идентифицируют количество значащих бит в последнем октете информационного поля кадра, четвертый бит равен единице, а последние 4 бита используются для нумерации пакетов в точке передачи от 1 до 15. Кадры индикации факсимильных характеристик отличаются от кадров начала факсимильного потока тем, что их девятый и десятый октеты содержат информацию о виде модуляции и скорости передачи факсимильного сообщения. В настоящее время изучается возможность использования протокола FADCOMP и для передачи факсимильных сообщений, передаваемых в соответствии с Рекомендациями МККТТ V.33 и V.17 (7200, 9600, 12000, 14400 бит/с).

(Окончание следует)


Этой статьей мы продолжаем рассмотрение мультиконсольных систем.

Этапы эволюции многопользовательских систем

Арсений Белоусов, Андрей Глазунов,
Максим Панкратов

Современный персональный компьютер для офиса — это совсем не та медлительная «пишущая машинка» с MDA-видеоадаптером, которая появилась в 1981 году. Сейчас солидные бизнесмены ставят на столы себе и своим сотрудникам мощные машины, сравнимые по производительности с рабочими станциями и миниЭВМ. Это, конечно, вполне в духе требований современного программного обеспечения. Но использование мощных компьютеров в роли однопользовательских пишущих машинок все-таки чрезмерно дорого. Мы полагаем, что сфера управления нуждается сегодня не столько в наращивании мощности процессоров, сколько в удобстве доступа к информации по приемлемым ценам.

Разумным выходом из создавшегося положения представляется использование так называемых мультиконсольных систем. Такие системы основаны на применении технологий, позволяющих оборудовать мощные компьютеры (с большим объемом памяти и процессором не ниже 386) дополнительными рабочими местами. Характерной особенностью подобных систем является легкость их включения в качестве фрагментов в локальные сети с различными протоколами. Основными сферами применения мультиконсольных систем следует считать офисную автоматизацию, сферы обучения и торговли и тому подобные.

Многопользовательские системы в России

Здесь рассматриваются системы, позволяющие расширить персональный компьютер как минимум на четыре рабочих места, снабженных графическим монитором, клавиатурой и дополнительными устройствами (мыши, принтеры, плоттеры, модемы, сканеры и т.д.). В настоящее время в России доступны мультиконсольные решения трех фирм, перечисленные в порядке убывания их стоимости: SunRiver (системы ST-SVGA и ST-UVGA), Maxpeed (SuperVGA MaxStation) и Advanced Micro Research (система UnTerminal) (табл. 1). В базовых конфигурациях вышеперечисленные системы дешевле X-терминалов и сетевых решений. Системы во многом аналогичны по возможностям и производительности и конкурируют, в основном, по ценам и надежности. Операции по установке всех систем вполне элементарны.

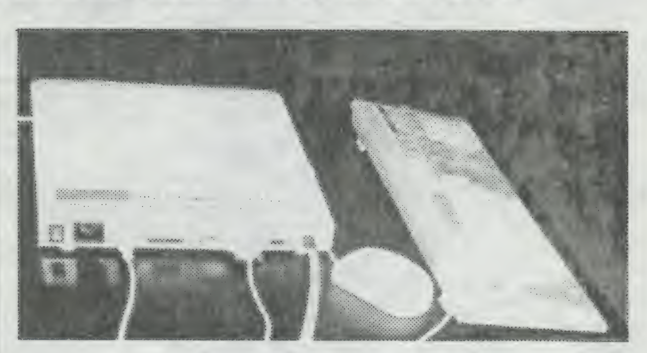


Рис. 1

Типичный комплект на четыре рабочих места состоит из платы, вставляемой в слот расширения материнской платы компьютера, кабелей и интерфейс-

Таблица 1

Краткие характеристики SunRiver ST-SVGA и ST-UVTGA	
Размеры базового блока	308×257×43 мм
Объем видеопамяти	1 Мбайт (ST-SVGA) и 2 Мбайт (ST-UVGA)
Климатические требования	рабочая температура от 0 до 50°С температура хранения от -40 до 60°
Электропитание	переменный ток 120/240 В
Краткие характеристики SuperVGA MaxStation	
Размеры базового блока	318×305×38 мм
Объем видеопамяти	512 Кбайт
Климатические требования	рабочая температура от 0 до 40°С температура хранения от -40 до 60°
Электропитание	переменный ток 110/220 В
Краткие характеристики UnTerminal VGA	
Размеры базового блока	300×330×35 или 190×90×35 мм
Объем видеопамяти	256 Кбайт, 512 Кбайт, 1 Мбайт
Климатические требования	рабочая температура от 0 до 40°С температура хранения от -40 до 60°
Электропитание	переменный ток 110/220 В

ных устройств. Сердце многопользовательской системы — плата ввода-вывода, содержащая микросхемы видеоадаптеров и другие интерфейсы, другими словами — полные консоли, достаточные для поддержания четырех рабочих мест. Пользователь на своем столе имеет дело с небольшим устройством, к которому подключаются дисплей, клавиатура, мышь, принтер и прочее. На рис. 1 и 2 изображены две модификации терминалов UnTerminal фирмы Advanced Micro Research, на рис. 3 — полностью собранный пользовательский терминал, за которым работают, «как за персональным компьютером». Ирония судьбы в том, что раньше приходилось объяснять, что персоналка — это ЭВМ, рассчитанная на один пользовательский терминал, а теперь ситуация диаметрально противоположная.

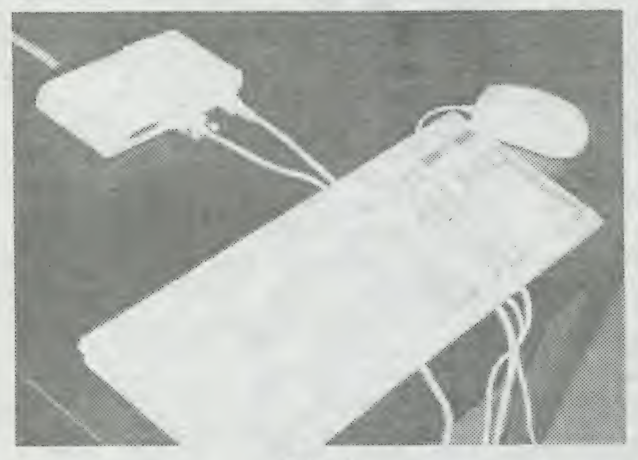


Рис. 2

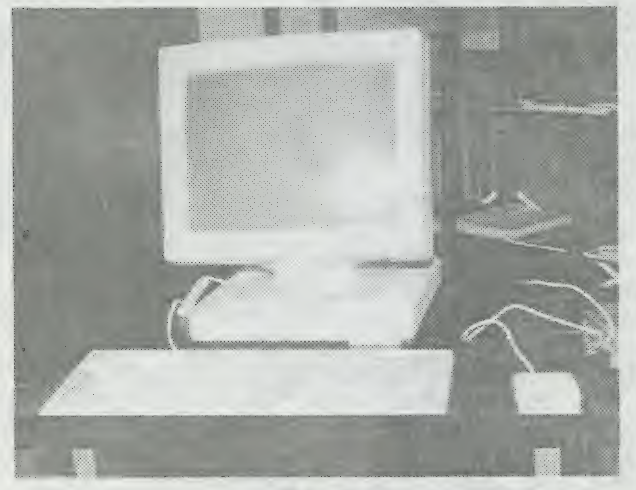


Рис. 3

Для работы нескольких терминалов, разумеется, необходима многопользовательская операционная система. В табл. 2 указаны многопользовательские операционные системы, поддерживающие мультиконсольное оборудование перечисленных компаний.

Таблица 2

SunRiver	SCO	UNIX, XENIX, Open Desktop
	Interactive	UNIX, X11
	Citrix	Multiuser 2.0
	IGC	VM/386
Maxpeed	SCO	UNIX, XENIX
	Novell	UnixWare
	CCI	Concurrent DOS, Multiuser DOS
AMR	SCO	UNIX, XENIX, Open Desktop
	Interactive	UNIX
	GC	VM/386
	CCI	Concurrent DOS
	Microbase	VirtuOS/386

В основном для работы с многопользовательскими системами применяют те или иные модификации UNIX. Однако в условиях широчайшего распространения программного обеспечения для PC, ориентированного на DOS и Windows, особенно на отечественном рынке, приходится ориентироваться преимущественно на системы, совместимые с DOS. В дальнейшем мы более детально остановимся на многопользовательских операционных системах.



Сегодня мы познакомим читателей с новыми продуктами фирмы RAD Data Communications, являющейся разработчиком и производителем большого ассортимента высококачественной продукции для локальных и региональных сетей. RAD является членом RAD Group — мирового лидера в производстве продукции для сетей и сетевой интеграции. В России фирму RAD Data Communications представляет системный интегратор UNI Inc.

Новые продукты фирмы RAD

Камилл Ахметов

▲Переключатель SPS-3

SPS-3 — это миниатюрный трехпортовый многопротокольный сетевой переключатель. Он предназначен для небольших компаний и офисов. SPS-3 значительно дешевле полноразмерного переключателя. При этом он поддерживает скорость передачи данных до 64 Кбайт/с, каждый из трех синхронных портов может работать с протоколами X.25, Frame Relay, HDLC, SNA/SDLC, собственным протоколом фирмы RAD — STM, и позволяет вести асинхронную передачу.



Благодаря SPS-3 нет необходимости подвергать мосты и маршрутизаторы модернизации для получения доступа к сервису X.25 или Frame Relay. Многопротокольный переключатель обеспечивает прозрачную HDLC-передачу через сети X.25 и Frame Relay. Также SPS-3 позволяет осуществить доступ к сети Frame Relay при помощи моста или маршрутизатора, имеющего интерфейс X.25.

SPS-3 предусматривает и рост сети. Если один из портов переключателя присоединить к статистическому концентратору RAD STM, он сможет поддерживать до 24 асинхронных каналов. SPS-3 — представитель семейства многопротокольных пакетных переключателей RAD. Эти переключатели поддерживают от одного до двенадцати синхронных портов и от двух до двадцати четырех асинхронных каналов.

▲RAD APS-8, APS-16, APS-24

Это устройства сетевого доступа по протоколам X.25 и Frame Relay. RAD выпускает 8-канальный APS-8, 16-канальный APS-16 и 24-канальный APS-24.

Устройства APS оперируют тремя соединениями, каждое из которых может независимо конфигурироваться для X.25, Frame Relay, HDLC, SDLC (QLLC), STM и асинхронной передачи. В конфигурации X.25 APS может поддерживать как постоянные, так и переключаемые виртуальные схемы, работая с пакетами размером до 4 Кбайт. В соединении Frame Relay APS оперирует пакетами Frame Relay размером до 8 Кбайт. С поддержкой HDLC APS можно использовать для соединения мостов и маршрутизаторов сетей X.25 или Frame Relay.



Для приложений, требующих более двадцати четырех асинхронных каналов, или для растущих сетей RAD можно присоединить к APS статистический концентратор RAD STM. При этом STM автоматически станет расширением APS, и APS сможет поддерживать до двадцати четырех добавочных асинхронных каналов через любое из синхронных соединений.

По материалам фирмы RAD Data Communications

Именная техника в России — роскошь или норма?

Круглый стол с Анатолием Карачинским и Николаем Федуловым ведут Камилл Ахметов и Сергей Новосельцев

Одной из особенностей сегодняшнего компьютерного рынка России является приход на его территорию техники brand-name-фирм. Brand-name значит "с именем". После того, как к концу 80-х годов рынок был заполонен дешевой аппаратурой посредственного и откровенно плохого качества, обращение к именной технике — естественный этап.

В круглом столе, посвященном проблемам продажи brand-name в России и общим перспективам развития компьютерного бизнеса, мы пригласили участвовать Николая Федулова, председателя совета директоров компании RUI IMC Apple Computer, и Анатолия Карачинского, президента фирмы IBS, авторизованной как бизнес-партнер корпорации Dell Computer.

Наш выбор был не случаен. И Николай, и Анатолий работают на российском компьютерном рынке со времени его зарождения, оба много сделали для его формирования. Оба с самого начала сориентировались в своей деятельности именно на brand-name-технику — хотя тогда это вызывало недоуменные вопросы, ею практически никто не занимался. Оба известны и как аналитики, имеющие свои взгляды на развитие рынка и отрасли в целом. Интересно, что волею судьбы оба они сменили "профилирующую" компьютерную платформу — Анатолий в 1992 году перешел с Apple на Dell, а Николай, наоборот, с IBM PC-совместимых компьютеров на Apple, что придает их видению дополнительную объемность.

Несколько менее полный текст этой беседы опубликован в февральском приложении к журналу "Деловые люди".

КомпьютерПресс. Давайте сначала оговорим "правила игры". Нам — и, надеемся, читателям — было бы интереснее провести эту беседу с вами не как с представителями компаний, занимающихся продвижением определенных торговых марок в России, а просто — с Анатолием Карачинским и Николаем Федуловым, которые много лет работают на этом рынке и имеют свой взгляд на его проблемы. Мы бы хотели услышать личное мнение, ваше vision, которое может совпадать, а может и не совпадать с официальной позицией ваших фирм. Вы согласны?

Федулов. Да, я думаю, что так мы сможем быть более объективными. Я, собственно, в таком смысле и понимал с самого начала ваше приглашение.

Карачинский. Я не возражаю.

КомпьютерПресс. Начнем с общих проблем поставщиков brand-name-техники в России. И для начала зададим несколько провокационный вопрос. RUI представляет в России продукцию фирмы Apple Computer, а IBS работает как главный представитель Dell Computer — фирмы, поставляющей IBM PC-совместимую технику. По определению вы — конкуренты...

Федулов. При текущем состоянии российского рынка у нас значительно меньше оснований для конкуренции, чем, скажем, у Apple и Dell в США, в условиях развитого рынка. И даже наоборот, я считаю, что на данном этапе мы вообще не конкуренты, а едва ли не союзники, нас объединяет — должна объединять — борьба за цивилизованный рынок, за увеличение реальной доли brand-name-техники в России. И меня радуют успехи brand-name-фирм в России, будь то закупка техники Hewlett-Packard Государственной Думой или техники Dell Сбербанком. На сегодняшнем этапе любой такой успех играет на руку всем brand-name-фирмам на нашем рынке: все это увеличение доли цивилизованного рынка в стране, вытеснение no-name-техники.

Карачинский. Можно говорить о разной конкуренции, по разным критериям.

Например, о конкуренции между целыми платформами или между отдельными фирмами по сегментам рынка. По-моему, конкуренции между платформами IBM PC и Apple в России нет. По крайней мере, в ее классическом варианте. Я хорошо знаю ситуацию с Apple, поскольку сам представлял интересы этой фирмы, когда работал в Интермикро. И я очень хорошо отношусь к платформе Macintosh, это, конечно, замечательный компьютер.

Но при этом компьютеры Apple имеют в России достаточно слабые позиции, несмотря на активную работу RUI. Это связано исключительно с полити-



Николай Федулов

кой самой фирмы Apple Computer, которая очень сильно опоздала с выходом на российский рынок, что не позволило пока компьютерам Apple занять на этом рынке подобающее им место. До 1991 года они вообще не хотели обращать внимания на этот рынок, да и потом вели себя достаточно пассивно, как бы не развивая рынок, а только присутствуя на нем.

Я думаю, что конкуренция имеет место главным образом в идеологическом плане, в плане выбора направления дальнейшего развития. То есть Apple сегодня не участвует здесь в конкуренции за масштабы рынка. С другой стороны, у Apple есть замечательные ниши, которые она заняла — не могла не занять.

КомпьютерПресс. А в чем, собственно, состоят проблемы с продажей brand-name в России? Как развивается этот бизнес, кто раньше занимался продажей компьютерной техники и чем был в основном заполнен рынок?

Карачинский. Сейчас рынок нормально эволюционирует, появление brand-name — просто закономерный элемент этой эволюции. Эволюции, связанной в большей степени с изменением психологии наших людей. Эта психология за последние десятилетия все-таки довольно сильно пострадала. Конкуренция, которая на Западе всегда являлась основой, двигателем любого товарного рынка, здесь отсутствовала. Все привыкли к тому, что у всех все, условно говоря, одинаковое: одинаковые машины, квартиры, зарплаты... Не было никакой градации жизни, не нужно было ничего выбирать. Что такое brand-name? Это престижная, проверенная, заслуженная марка, о которой все известно и которая не может допустить снижения своего уровня. Если на Западе принято покупать дома, определенные марки автомобилей или компьютеров, соответствующие вашему социальному статусу или престижу фирмы — то здесь основным критерием была цена, а такие категории, как престиж, практически отсутствовали. И вот когда в России рынок наконец появился, все первое время продолжали существовать на нем так же, как и до него. Считалось, что все как бы одинаковое и главное — купить подешевле. И побольше. Менталитет, который закладывался в нас довольно долго, сработал.

Возвращаясь к специфике компьютерного рынка, нужно еще учесть, что круг компьютерщиков, разбирающихся в проблемах, был достаточно узок, и он слабо пересекался с кругом лиц, принимающих решение о закупке. Покупателями в первой волне этого рынка были, в основном государственные организации, и закупки компьютеров производились по каким-то плановым показателям, а не потому, что они были нужны. Так что вопросы работы, сопровождения закупленной техники недостаточно сильно волновали тех, кто распоряжался деньгами. Гораздо сильнее их волновало стремление потратить эти деньги "разумно", чтобы их не обвинили, что они купили компьютеры в два раза дороже, чем предлагала какая-то еще фирма. Они жили сегодняшним днем, и что будет завтра с купленными компьютерами их не очень интересовало. Все это привело к тому, что процветал рынок дешевой, не именной техники.

Появление brand-name связано с началом формирования новой психологии. Мы поняли, что качество очень сильно зависит от цены. Если раньше думали только о том, чтобы купить побольше и заплатить поменьше, то теперь мы стараемся вложить деньги наиболее эффективно, тем более, что заметно увеличилась доля частных денег. Появилось понятие престижа. Наконец, очень велико воздействие рекламы. Успех фирм, поставляющих даже самую качественную технику, во многом зависит от маркетинга и рекламы. Заметьте, что реклама brand-name обычно тоже существенно другая, она качественнее, она по другому построена. У них профессиональные маркетинговые команды, которые занимаются созданием имиджа, продвижением имени фирмы. И вот под воздействием этих факторов маятник качнулся в другую сторону, в сторону brand. Может быть, даже несколько больше, чем было бы в другой ситуации, при "нормальном" развитии, в стране с таким экономическим положением и уровнем развития технологии. Так что я считаю, что проблемы brand-name нет. Есть естественная эволюция рынка в наших специфических условиях.

Федулов. Сначала небольшой комментарий к самому понятию. Понимаете, слово "brand" довольно неоднозначно. Когда мы говорим "brand", мы обычно подразумеваем IBM, Compaq, Hewlett-Packard, Dell, Apple и так далее. Но можно говорить о brand и как о маркетинговом феномене. Вот, скажем, компьютеры марки "Лэнд". Я не знаю качества этих компьютеров, я на них никогда не работал. Но я вижу явную попытку сделать brand: налицо достаточно серьезные маркетинговые усилия, которые вложены в марку "Лэнд", в это втянуты достаточно профессиональные команды. И шансы сделать brand-name вроде были — если бы не еще одна наша особенность, тоже из области психологии и не имеющая прямого отношения к компьютерам. Это сложилось исторически, тянется, может быть, еще с Петра: все, что не наше — чем-то лучше. Для нас самые дорогие и

"накрученные" "Жигули" остаются хуже весьма средней "Ауди".

Или возьмем Dendy — это brand или нет? Я считаю, что Dendy — это brand-name, сделанный в России. Стиплер — профессиональная команда, они подключили деньги, они проработали идею, маркетинговые методы, понимание того, как "сделать" этот рынок, — и "сделали" рынок Dendy. С этой точки зрения, насколько бы устаревшими решениями не были обеспечены все эти Dendy — это рынок! Это действительный brand. И заметьте, сейчас предпринимаются попытки повторить этот успех, сделать brand с нуля: Kenga, Amata, и KAMI что-то свое начинает двигать... Поэтому в широком смысле brand — это все, что получило имидж интересного продукта, то, что рынок — нормальный рынок — потребовал.

Если же говорить о компьютерном рынке, об истории "брэндинга" здесь — я бы выделил три этапа продвижения персональных компьютеров на наш рынок.

Первая волна закупок техники началась в 1983 году. Как ни странно, это был период интереса к технике brand-name. Я тогда работал в Торгово-Промышленной Палате и был хорошо информирован о производимых закупках. Это были IBM, Hewlett-Packard, DEC, Wang, Olivetti и другие, всего 12-15 фирм. Рынка как такового в стране еще не было, и интерес к персоналкам в значительной степени искусственно поддерживался этими фирмами. Они еще и потому делали здесь упор на персональные компьютеры, что не могли напрямую поставлять более мощную технику, которая была под эмбарго. А для СССР персоналки были чем-то совсем новым, работали все тогда на мэйнфреймах и мини-компьютерах. Информации о ПК почти не было, журналов о персоналках и так-то было мало, а сюда они практически не доходили. Фирмы же эти работали здесь давно, пустили какие-то корни, от них и шла значительная часть информации. (В то же время я знал целый ряд no-name-фирм, которые в эти годы очень старались выйти на наш рынок, но так никогда и не были востребованы и исчезли.) А потом пошла мода на персоналки, и все сидели и считали: вот он, мегабайт! Памяти больше, чем у нас в IBM/360!

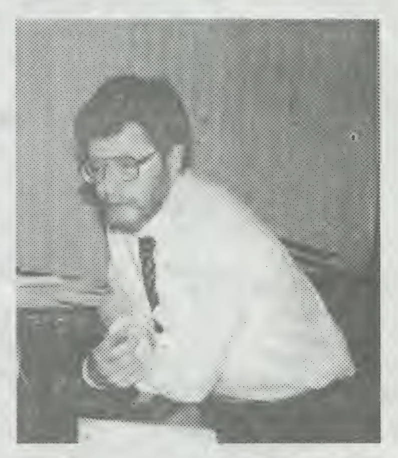
Итак, на первом этапе превалировали компьютеры brand, стоили они тогда бешеных денег, но и закупались в небольших количествах — чтобы посмотреть и понять, что с этим делать.

Начало второй стадии относится к 1985-1987 годам, когда уже в массовом количестве появилась noname-техника, та самая, которую можно было брать "побольше и подешевле", экономя государственные деньги. Сначала это навязывалось Элоргом, через который шло большинство закупок. У него были отработаны технология закупки по 10-12 фирмам, конъюнктурные проработки, шаблоны... Покупателя отправляли в фирмы 1, 2, 3, заранее зная, где он купит. У Элорга не было понятия brand, они как бы отвечали за экономию государственных средств. Постепенно в ходе перестройки доминанта Элорга исчезла, но... Компьютер, который раньше был товаром только для Элорга, занимавшегося закупками от имени государства, — стал товаром для всех, кто решил себя попробовать в этом новом бизнесе, товаром, на котором можно заработать нормальные деньги. Если сейчас посмотреть назад, то окажется что очень значительный процент известных в дальнейшем компаний и структур начинали с этого "торгово-посреднического" бизнеса, занимались им. "Диалог", кооператив Тарасова, та же МММ... Заводы скидывали деньги, закупая большие партии компьютеров, а потом сами начинали их понемногу перепродавать. Об именной технике речь по-прежнему не шла, а критерий Элорга — экономия госсредств — трансформировался в стремление купить подешевле, продать подороже. Так что нашествию неизвестной техники способствовала не только психология покупателей, о которой говорил Толя, но и поведение продавцов. Таков был второй этап, этап, когда появился товар, но не было рынка. Истинный рынок еще не сформировался, не было осознанного запроса, не было понятия маркетинга.

А потом наступил третий этап, расцвет которого мы сейчас переживаем.

Разумеется, дешевая техника никуда не делась, но, как уже говорилось, появились покупатели, желающие правильно вложить свои деньги, купить надежную технику — или хотя бы просто не такую, как у всех. Появились целые структуры, у которых были хозяева, желающие купить за свои деньги что-то приличное. Вот это уже был рынок, поскольку появилась настоящая альтернатива. Где-то в 1992 году резко вошли в моду компьютеры от Compaq, Toshiba, ALR, AST Research. Еще в конце 1991 мы в Merisel (а мы были чуть ли не первыми, кто получил свободный доступ едва ли не ко всем brand) пытались их продвигать — но они не шли! Так что перелом я отношу где-то к середине 92-го года.

Карачинский. Когда в 1991 году мы в Интермикро взялись продавать компьютеры Apple, над нами поначалу просто все смеялись — такая дороговизна, кому это нужно! То же было,


Анатолий Карачинский

кстати, когда уже в IBS мы стали работать с Dell. Браться за такую дорогую технику, когда все вокруг покупали и продавали компьютеры партиями 5-10 тысяч штук по каким-то безумно низким оптовым ценам, по 700 долларов что-ли... Многим это казалось безнадежной затеей. И все-таки в 1992-1993 годах кооператоры с огромными партиями дешевых компьютеров исчезли, и все больше компаний стали работать с brand-name.

На рынке, действительно, очень многие стараются повторить успех других компаний, рынок — это очень обезьянья вещь. И ситуация с brand тоже подчиняется этому правилу. Как только кто-то начал продавать brand-name-технику — за ним последовали другие. Помните, Стиплер начал продавать лазерные принтеры Hewlett-Packard? Тот, первый, Стиплер вообще был классической командой — первой частной, негосударственной компанией, которая смогла классически раскрутить рынок — за что теперь ей все благодарны. И сразу появилось очень много фирм, которые продают принтеры Hewlett-Packard.

Федулов. Мне кажется, что именно Hewlett-Packard начал серьезное продвижение brand-техники в Россию. В 1991 году фирма Merisel-KT стала первым дистрибьютором Hewlett-Packard по линии персональных компьютеров (а Стиплер тогда же подписал договор по принтерам), причем мы во многом подписывали договор из протокольных соображений, это как бы было полезно и нам, и им. Но спроса тогда не было совсем, и цены были совершенно невероятные, нельзя было по таким ценам ничего продать. Однако за год серией снижений цен компьютеры Hewlett-Packard стали brand-name в России, и сегодня является здесь либо первой, либо второй компьютерной фирмой! Этого нет в других странах, там он не входит и в десятку: HP — традиционно brand, но на рынке принтеров, периферии. Так что я считаю, что у этого успеха две составляющие — с одной стороны, имя Hewlett-Packard, с другой — очень грамотное снижение цен.

Карачинский. Но имя Hewlett-Packard на потребительском рынке сделал все-таки Стиплер. Они начали рекламную кампанию в широкой прессе, это была первая настоящая маркетинговая public-стратегия. А сама фирма Hewlett-Packard дала здесь первую рекламу только полтора года назад... Я считаю, что на компьютеры Hewlett-Packard многие дилеры обратили внимание именно потому, что надо было противопоставить какой-то brand компьютерам Dell, которые начали активно продвигаться на рынке. А благоприятная ситуация с компьютерами Dell в России обусловлена тем, что здесь их эксклюзивно продвигает российская фирма, а не собственно Dell, иначе бы их реально просто никто не продвигал.

Или, например, Compaq — эта компания фактически пришла сюда третьей год назад или чуть больше — и, по моим оценкам, по объемам они сегодня тоже третьи, причем, если они своей политики не изменят, то так и останутся третьими. Им удалось сделать несколько крупных контрактов — но на рынке они пока не очень видны. У Compaq в России больше десятка дистрибьюторов, но нет дистрибьютора, который бы занимался только техникой Compaq, Compaq как brand фактически никто не продвигает. Фирма должна или сама начать активно продвигать себя (как это стал сейчас делать Hewlett), или же назначить, условно говоря, уполномоченного дистрибьютора, перед которым поставить эту задачу. Иначе получается ситуация, когда никто из дистрибьюторов не хочет вкладывать деньги, делать большую работу, плодами которой воспользуются их конкуренты, которые тоже продают Compaq! Посмотрите, кто сегодня из "больших" двигает Compaq?

Федулов. Пожалуй, активнее других Croc Incorporated — хотя их нельзя назвать дистрибьюторами, они скорее относятся к категории интеграторов.

Карачинский. Да, это VAR, value-added reseller, они продвигают свои решения на базе компьютеров Compaq. Но кто-то же должен построить технологию движения компьютеров Compaq — как и любой другой фирмы — на этот рынок, условно говоря, конвейер, по которому они сюда едут, а здесь сбрасываются, распределяются и т.п.

Федулов. Здесь я согласен с Толей. И тем не менее Compaq все же стал третьим — а может быть и вторым. Это тоже говорит о готовности нашего рынка к "брендизму" вообще.

Карачинский. Рынок, безусловно, готов, но "третий" в данном случае означает "последний", потому что больше никто пока в Россию не пришел.

Федулов. Почему же? Пришел, к примеру, Acer. И, думаю, Acer настроен — хотя и другими методами, с точки зрения, скажем, каналов — повторить здесь успех Dell.

Карачинский. Acer для меня все равно пока несколько в другой весовой категории. Им предстоит работать, чтобы стать brand-name в полном смысле слова, возможно, они добьются этого в 1995 году. Кстати, мне очень нравится их маркетинговая стратегия. С другой стороны, не надо забывать разницу между IBS и Acer — по масштабам, по финансовым ресурсам, по возможностям. Мы, конечно, достигли некоторых успехов, но все же остаемся маленькой российской компанией, хотя и относимся к тому не очень большому числу компаний, которые реально влияют на формирующийся здесь рынок.

Федулов. Хотя мы и отходим несколько в строну, я скажу, что у вас уникальная ситуация на здешнем рынке. IBS, с одной стороны, дистрибьютор, эксклюзивно представляющий Dell в России, с другой

стороны — вы независимы и можете, в принципе, свободно работать с кем угодно.

КомпьютерПресс. Интересно, а насколько вообще, по вашему, крупные западные компании заинтересованы в работе на российском рынке?

Карачинский. Да в общем-то... ни на сколько. За последние пять лет мы так или иначе поработали со всеми крупнейшими в компьютерной индустрии компаниями — и мое мнение, что на самом деле Россию в качестве серьезного рынка никто из них не рассматривает. Готов объяснить, почему это так. Возьмем, к примеру... ну, пусть опять Compaq. Он реально продает в мире технику примерно на 10 миллиардов долларов. В России они продали в прошлом году, ну пусть даже на 100 миллионов — все равно это не больше одного процента. Я уж не говорю про IBM, у которого общий годовой объем по всем направлениям миллиардов 50-60. И сразу становится понятно, что Россия для них — это что-то крайне далекое. Представители этих компаний могут вести себя здесь как угодно круто и давать любые обещания, но надо же понимать, что в иерархии своей компании они занимают достаточно низкий пост, так что реально они не в состоянии решить ни одного вопроса. Они настолько далеки по этой иерархии от руководства компании, что скорее всего, их мнения там никогда и не услышат. А если и услышат, первым делом спросят — "а сколько мы там зарабатываем? Что? И вы предлагаете инвестировать туда такие деньги?!" И отправят вас к финансистам. А финансист — бог любой большой компании. Финансисты здесь у нас не бывают, и вообще ничего про эту страну не знают, кроме ее положения на карте. Они питаются простыми вещами: им приходят различные анализы, им приходят различные риски, им банки дают списки... И когда возникает вопрос, скажем, о кредите, финансист смотрит, что в списке финансовых рисков Россия находится на предпоследнем месте (перед Ираком или Ираном, точно не помню). А он отвечает за то, чтобы его компания не потеряла деньги. Так что можно перед ним расписывать любые перспективы российского рынка, хоть плясать — он руководствуется тем, что дает его банк, а банк пользуется аналитическими нью-йоркскими или там лондонскими разработками. Все. Он говорит вам "нет" — он не имеет права поступить иначе.

Так что работать в России с западными компаниями очень трудно. Вы хотите сделать свою работу гибкой, приспособить ее к условиям этого рынка, наладить поставки, создать склад... Вам говорят: "Да, все это очень хорошо", — а потом ничего не происходит. Все, кто начинает работать с ними, с этим сталкиваются, так что надо быть готовым... А вот в Китае, где степень финансовых рисков мала, где государство дало гарантии почти везде, активность западных компаний очень велика, и они добились там больших успехов.

Федулов. Год назад я был бы с тобой согласен на 120%. Но сейчас начинает кое-что меняться. И меняют ситуацию в первую очередь "азиатские драконы" — Acer, Samsung. Посмотри, насколько они активны в этом году. На западноевропейском и американском рынке шансов у них практически нет, и потому к российскому они отнеслись серьезно. Тут, по-моему, показателен пример Мексики. Эта страна тоже не была приоритетным рынком для американских фирм. В итоге мексиканский рынок просто потерян для Штатов, потому что им фактически завладел Acer. И мне кажется, что через год, когда американскими фирмами будут осознаны действия Acer здесь, это должно вызвать определенный резонанс, они тоже подкорректируют российскую политику. Они все-таки реагируют на изменение ситуа-

ции — обрати внимание, как оживился здесь Hewlett-Packard после неожиданно больших, непрогнозируемых объемов продаж. И фактор чужого примера, о котором мы говорили применительно к нашим фирмам, справедлив и для западных.

Карачинский. Ты чересчур оптимистичен. Да, те люди, которые работают здесь, могут отслеживать, к примеру, действия Acer'a, адекватно реагировать, но как только они выходят с этим на следующие ступени иерархии, вступает в действие тот механизм, о котором мы говорили. Даже если они смогут пробиться на высшие уровни, для того чтобы что-то существенно поменять, надо менять такие святыни, как, например, Корпоративная Стратегия, принятая Советом Директоров. Большие фирмы очень бюрократизированы. Это одна из причин, почему мы работаем именно с Dell — фирма молодая и пока еще более гибкая, чем скажем IBM или Compaq. Сейчас мы пытаемся работать с IBM, но это тяжело.

Федулов. Я согласен, что в ближайшее время Россия не будет для ведущих производителей ПК приоритетным рынком. И IBM не прибежит сюда и не скажет: вот вам 100 миллионов, раскручивайте этот рынок. Это, однако, не означает, что Россия вообще не будет рынком для них. Они будут работать на нашем рынке, и их доля на нем будет расти. Да, расти не за счет усилий этих фирм — а благодаря действиям российских компаний, их здесь представляющих. Dell, Hewlett-Packard и Compaq, конечно, обязаны своими успехами здесь тому, что этим занялись именно российские фирмы.

Карачинский. И ни Gateway, ни AST, ни Packard Bell не предпримут никаких действий по отношению к России, пока здесь не появится местная компания, которая будет их продвигать.

Федулов. Но давай взглянем и с другой стороны. Все таки 100 миллионов в год для России — это 10 процентов ее компьютерного рынка. Значит, такова — очень грубо — доля Compaq на этом рынке. Добавим сюда показатели HP, Dell, Apple, IBM — и мы увидим, что суммарная доля brand-name-компьютеров на этом рынке вскоре может перевалить некую критическую массу...

Карачинский. Может быть, это уже в известном смысле произошло.

КомпьютерПресс. А как вы расцениваете заявление IBM о прекращении производства компьютеров на процессоре Pentium?

Карачинский. По-моему, это больше политика. К технологии, к реальной жизни это имеет слабое отношение.

Федулов. Можно было бы усмотреть в этом шаг IBM к смене платформы (хотя я не склонен к такому упрощенному толкованию). IBM, как и Apple, разумеется, заинтересована в продвижении процессора PowerPC.

Карачинский. Я не думаю, что PowerPC вытеснит процессоры Intel. PowerPC, конечно, займет свое место на рынке, тем более что его двигают сильнейшие фирмы — IBM и Apple. Но Intel поддерживает большинство на рынке, а кроме того, у них колоссальный технологический задел.

Победа на рынке — это не только архитектура процессора. Объясните мне как клиенту, зачем мне переоборудовать мой офис на PowerPC?

Федулов. А как три года назад все спрашивали — зачем мне Windows?

Карачинский. Ну, программную платформу заменить очень легко. Это, знаете ли, совсем другие инвестиции.

С компьютерами Apple другая ситуация — их пока выпускает только фирма Apple, и раз она переходит на новый процессор, то выбора у клиента не остается.

Федулов. Соответственно, все 13 процентов мирового рынка персональных компьютеров, которые до сих пор работали на Apple Macintosh, станут пользоваться Power Macintosh, то есть PowerPC. Ну, а теперь открыта для лицензирования операционная система Macintosh, и через шесть-восемь месяцев на мировом рынке появятся клоны компьютеров Apple. Наконец, уже сейчас существует шесть операционных систем, заведомо работающих на процессоре PowerPC.

Карачинский. Ну и что? Ведь непонятно, какая из них станет стандартом. А может быть, и вовсе не установится стандарта, будут любители Windows, OS/2, Power-архитектуры и так далее. Когда IBM, Apple и Motorola взялись за PowerPC, они хотели еще раз изменить стандарт рынка. Теперь еще раз стало понятно, что взять и поменять рынок нельзя.

Федулов. Но переход и задуман не как резкий скачок, а как постепенное врастание нового стандарта — подчас незаметно, "прозрачно" для пользователя. Вспомни, как постепенно завоевывали рынок к примеру, стандарты на шину EISA или PCI. Сейчас просто рано еще подводить итоги, процесс только начался. Можно лишь улавливать какие-то тенденции, знаки будущих изменений... Однако для пользователей Макинтош этот переход, можно считать, уже свершился, почти все выпускаемые сегодня desktop-компьютеры и серверы Apple оснащены процессором PowerPC. И пользователи этого практически не заметили. Они покупают новый Mac — PowerMac — и для них внешне все остается как раньше: интерфейс, ОС, прикладные программы, платы расширения... И только внутри стоит другой, более быстрый процессор. Такова же стратегия перевода на PowerPC, на единую платформу Apple-IBM-Motorola, и пользователей с других платформ. Как мы уже говорили, практически все операционные системы уже работают на PowerPC — OS/2, Windows NT, Solaris, AIX,



NetWare. Разработчики основных прикладных программ — текстовых процессоров, электронных таблиц, баз данных, графических и мультимедиа-приложений — уже выпустили или работают над созданием версий своих продуктов для PowerPC. Платы расширения — в общем стандарте PCI и их не надо менять.

Карачинский. Но посмотри, даже в моей сравнительно небольшой компании специалисты сегодня никак не решат — нужно ли переходить от Novell NetWare на Windows NT. Потому что, хотя Microsoft и пыталась сделать так, чтобы такой переход был максимально простым, какие-то проблемы все равно возникают, какие-то приложения с ходу не идут. Вот у меня, например, сделана масса собственных специализированных прикладных программ. И мы их не решаемся перевести на новую ОС на том же процессоре. А если менять еще и процессор...

Федулов. Процессор PowerPC 615, разработанный IBM, выполняет и команды 486-го процессора, так что не надо менять ничего — ни ОС, ни программ. Потом, если исходные тексты сохранились, их можно просто перетранслировать. Если нет — они пойдут в режиме эмуляции. Но я согласен, вопрос с собственным программным обеспечением — самый трудный при переходе.

Что касается вопроса "а зачем?" — он возникает всегда, как возникал и при переходе с DOS на Windows. Сегодня этот переход состоялся, и Windows дают пользователю не только новый для пользователей этой платформы интерфейс, но и новую функциональность, массу возможностей, немыслимых в рамках DOS. Но для того чтобы переход стал возможен, нужен был более мощный процессор. На 286-м никаких Windows быть не могло. Именно в этом главный смысл и перехода к PowerPC: создать платформу, которая сможет обеспечить качественно новую функциональность компьютеров. И эти новые черты, новые системы появятся где-то в 1996-97 годах. Вот тогда и будет решаться вопрос о будущем той или иной платформы.

Карачинский. Но для того чтобы вытеснить Intel с рынка, заставить пользователя Intel перейти к PowerPC, нужны очень веские преимущества. И в эти годы Intel тоже не будет стоять на месте.

КомпьютерПресс. Напоследок хотелось бы услышать о том, как каждый из вас лично видит будущее рынка персональных компьютеров.

Федулов. Ну что ж... Бума PowerPC, конечно, в ближайшее время не будет. Будет постепенное внедрение на рынок, завоевание места под солнцем. Вообще 1995 будет "проходным" годом, фирмы будут определять стратегию на рынке, выбирать для себя линию на будущее — P6, P7, PowerPC... Но внешне, для пользователей, не будет никаких коренных изменений. Видимо, какое-то время Apple еще будет в одиночестве. Больше чем 15-20 процентов на рынке персональных компьютеров PowerPC пока, видимо, не получит.

Мне кажется, очень большую роль сыграют традиционно некомпьютерные фирмы — Sharp, Sony, Matsushita и др. Фирмы этого ряда уже сейчас чувствуют предел того рынка, в котором они существуют, рынка обычной, "тупой" домашней электроники. Тенденция "поумнения" бытовой электроники (теле-, видео-, аудио-, коммуникационной), ее сближения с компьютерами заметна уже сегодня, и она должна очень отчетливо проявиться в 1995 году — уже для конкретных потребителей. Наблюдается снижение стоимости вхождения для потребителя (я имею в виду не только денежные, но и интеллектуальные затраты) в эту "компьютерную" часть. И здесь наличие дешевого и мощного чипа типа PowerPC должно сыграть очень важную роль при выборе указанными фирмами спектра возможностей, закладываемых ими в следующие поколения своих устройств. Что будет дальше, в 1996 году, я пока не готов оценить, слишком сложна ситуация и нужно посмотреть, какие шаги будут предприняты в 1995.

Карачинский. Здесь я согласен с Колей, и считаю, что рынок consumer electronics может в перспективе похоронить рынок персональных компьютеров. Разнообразные бытовые электронные устройства и персональные коммуникаторы будут делать то же, что сейчас персоналки, и пользователи постепенно перейдут на них. Компьютеры же останутся для каких-то узкоспециальных, вертикальных применений — как это произошло с мэйнфреймами, которые были вытеснены персональными компьютерами. Рынок ПК будет совсем небольшим, но рядом будет огромный рынок всевозможных интеллектуальных бытовых устройств, которые умеют все. И дети будут пользоваться одними, хозяйки — другими, бизнесмены — третьими. У всех будут свои интеллектуальные помощники. А если они еще научатся понимать голос... Это будет что-то вроде того, о чем мы когда-то читали в фантастических романах.

Предлагаем вниманию читателей в 1995 году:

А.Борзенко, А.Федоров. Мультимедиа для всех

Книга в популярной форме рассказывает о последних новинках компьютерной индустрии в области мультимедиа: о звуковых картах, приводах компакт-дисков, джойстиках, о том, для чего все это нужно и как с этим обращаться. Предназначена для тех, кто ничего не слышал о мультимедиа, кто слышал, но не знает или не понимает, зачем это нужно. Словом, это книга о том, как просто и недорого установить средства мультимедиа на своем персональном компьютере.

К.Ахметов. Курс молодого бойца

В виде учебного курса систематизированы материалы, представляющие собой расширение известной рубрики журнала, а также ранее не публиковавшиеся. Книга предназначена для обучения работе на IBM PC-совместимом компьютере в средах MS-DOS и Microsoft Windows людей, не имеющих предварительной компьютерной подготовки.

Издание второе, расширенное и дополненное.

К.Ахметов. Microsoft Windows 95

Первое русскоязычное издание, посвященное работе пользователя с новой операционной системой Windows 95 фирмы Microsoft. Содержит описание интерфейса Windows 95, программных средств, входящих в состав системы, принципов работы с системой. Даны рекомендации для пользователей MS-DOS и Microsoft Windows.

К.Ахметов, А.Борзенко. Современный персональный компьютер

Книга содержит систематизированную информацию о современном программном и аппаратном обеспечении IBM PC-совместимых компьютеров. В доступной и увлекательной форме приведены сведения об архитектуре IBM PC, новейших устройствах ввода-вывода, модернизации компьютеров, работе в операционных системах MS-DOS, Microsoft Windows и IBM OS/2.

Для читателей, имеющих базовую компьютерную подготовку.

ДЛЯ ПОДПИСЧИКОВ КОМПЬЮТЕРПРЕСС

скидка 30%

К.Ахметов, А.Борзенко. Мультимедиа у вас дома

Персональные компьютеры стремительно входят в нашу повседневную жизнь. Но трудно представить себе домашний компьютер без средств мультимедиа. Эта книга поможет вам установить звуковую плату, дисковод для компакт-дисков, сделать правильный выбор программных средств и научит эффективно пользоваться ими.

А.Федоров. Создание Windows-приложений в среде Delphi

Книга содержит обширную информацию по новому программному продукту фирмы Borland — **Delphi**. В ней содержится описание интегрированной среды разработчика, визуальных компонентов, техники программирования и использования среды Delphi для разработки Windows-программ. Особое внимание уделено практическому программированию — из 20 глав, составляющих данную книгу, 16 посвящены созданию Windows-программ. В приложение вынесена справочная информация, дополняющая стандартную документацию.

Рекомендуется для самостоятельного обучения, в качестве справочного пособия, для широкого круга программистов, желающих быстро и эффективно создавать Windows-приложения.

Напоминаем нашим читателям, что многие подписчики КомпьютерПресс уже читают книги, приобретенные в редакции по цене вдвое меньше магазинной, и играют в полученные бесплатно и купленные с 40% скидкой замечательные игры фирмы НИКИТА, причем число играющих после выхода мартовского номера журнала возрастет ровно на число подписавшихся, а именно более чем на 20 тысяч человек.

Подписку на КомпьютерПресс Вы можете оформить

По каталогу РОСПЕЧАТИ — индекс 73217	**По каталогу Агентства «Книга-Сервис».** *Телефоны:* (095) 124-94-49, 129-29-09, 129-72-12. *Адрес:* 117168 Москва, ул. Кржижановского, д.14, корп.1	**По каталогу НТЦ Информ и Агентства «Деловая Россия» с рассылкой по территории России и стран ближнего зарубежья** *Телефоны:* (095) 129-68-29, 158-45-73	**В редакции КомпьютерПресс (без почтовой доставки).** *Адрес:* Москва, ул. Ленская, д.2/21 *Телефон:* (095) 471-32-63

Представляем самые быстрые в мире рабочие станции среднего класса. Семейство Alphastation

ALPHA™ GENERATION

Рабочие станции нашего нового 64-разрядного семейства Alphastation существенно превосходят скорости 32-разрядных машин. Три мощные системы семейства имеют тактовую частоту 166 МГц и выше. Этого более чем достаточно для решения самых серьезных вычислительных задач, в частности задач двухмерной и трехмерной графики, САПР-механики, САПР-электроники, а также множество других. Это главное, что позволило нашим рабочим станциям стать лучшими на рынке по соотношению цены и производительности.

Естественно, каждая Alphastation работает под управлением операционных систем UNIX, Windows NT и Open VMS, а также обеспечивает открытость архитектуры благодаря новой стандартной для промышленности локальной шине PCI. И если для Вас важны надежность и низкие эксплуатационные расходы, Вы по достоинству оцените трехлетнюю гарантию для наших рабочих станций.

Что касается общей ценности, то познакомьтесь с техническими данным и моделей семейства. Они говорят красноречивее всяких слов. Вы видите, что наши рабочие станции – не просто самые быстродействующие в мире, но и исключительно мощные во всех других отношениях системы.

И при этом вполне доступные по ценам.

	Alphastation 200 4/166	Alphastation 200 4/233	Alphastation 400 4/233
Корпус	Настольный	Настольный	Мини-башня
Тактовая частота процессора	166 МГц	233 МГц	233 МГц
SPECint 92	207,7	150,1	150,1
SPECfp 92	134,8	183,9	183,9
Число гнезд ВВ (PCI/ISA)	3	3	6
Макс. проп. способность ВВ	132 Мбайт/с	132 Мбайт/с	132 Мбайт/с
Макс. емкость ОЗУ	192 Мбайт	192 Мбайт	192 Мбайт
Макс. емкость внутр. массовой памяти	4,2 Гбайт	4,2 Гбайт	4,2 Гбайт

Представительства:
в Москве тел.: (095) 244-9540
в С.-Петербуре тел.: (812) 119-1040
в Киеве тел.: (044) 295-6910, 295-1431
факс: (044) 293-6041

